




Source classification in deep radio surveys using machine learning techniques

Zafiirah Banon Hosenie

 [orcid.org 0000-0001-6534-593X](https://orcid.org/0000-0001-6534-593X)

Dissertation submitted in partial fulfilment of the requirements for the degree

Master of Science in Astrophysics and Space Science

at the

North-West University

Supervisor: Dr N Oozeer

Co-supervisor: Prof. B Bassett

Assistant Supervisor: Prof. I Loubser

Graduation May 2018

29770610

Declaration of Authorship

I, Zafiirah Banon Hosenie, know the meaning of plagiarism and declare that all of the work in the dissertation titled “Source classification in deep radio surveys using machine learning techniques”, save for that which is properly acknowledged, is my own.

Contents

| | |
|---|-------------|
| Declaration of Authorship | i |
| Contents | ii |
| List of Figures | vi |
| List of Tables | ix |
| Acknowledgement | x |
| Abstract | xi |
| Ongoing Publications | xiii |
| Abbreviations | xiv |
| List of symbols | xv |
| 1 Introduction | 1 |
| 1.1 A brief history of Radio Astronomy | 1 |
| 1.2 Astronomical Sources | 2 |
| 1.3 Galaxies | 3 |
| 1.3.1 Active Galactic Nuclei | 3 |
| 1.3.2 Radio Galaxies | 4 |
| 1.3.3 Morphology and structure of radio galaxies | 4 |
| 1.3.4 Morphological Classification of Radio sources | 6 |
| 1.3.4.1 Fanaroff-Riley Class I (FRI) | 6 |
| 1.3.4.2 Fanaroff-Riley Class II (FRII) | 7 |
| 1.3.4.3 Differences between FRI and FRII | 8 |
| 1.4 Review of some source detection techniques | 9 |
| 1.4.1 Image Transformation | 9 |

| | | |
|----------|---|-----------|
| 1.4.2 | Detection Criterion | 10 |
| 1.4.2.1 | Thresholding | 10 |
| 1.4.2.2 | Local Peak Search | 11 |
| 1.5 | Surveys | 12 |
| 1.5.1 | Radio Surveys | 12 |
| 1.5.2 | SUMSS | 12 |
| 1.5.3 | The FIRST Survey | 14 |
| 1.5.4 | The NVSS Survey | 15 |
| 1.6 | Sample Selection | 16 |
| 1.6.1 | Point and Extended Datasets | 17 |
| 1.6.2 | FRI and FR II Datasets | 18 |
| 1.7 | Summary | 18 |
| 1.8 | Objectives | 19 |
| 1.9 | Overview of the thesis | 19 |
| 2 | Introduction to Machine Learning | 21 |
| 2.1 | Style of learning | 22 |
| 2.1.1 | Supervised Learning | 23 |
| 2.1.2 | Unsupervised learning | 23 |
| 2.2 | Machine Learning Algorithms | 23 |
| 2.2.1 | k Nearest Neighbours | 24 |
| 2.2.2 | Random Forest | 26 |
| 2.2.3 | Naive Bayes Classifier | 27 |
| 2.2.4 | Multi Layer Perceptron | 28 |
| 2.2.4.1 | The Architecture of the Multi Layer Perceptron | 29 |
| 2.3 | Dimensionality Reduction | 30 |
| 2.4 | Application of Machine Learning in Astronomy | 31 |
| 2.5 | Summary | 33 |
| 3 | Astronomical Source Detection using Filter-based Methods | 34 |
| 3.1 | Local Peak detection | 34 |
| 3.2 | Source Properties | 36 |
| 3.3 | Centroids | 37 |
| 3.3.1 | Image Moments | 37 |

| | | |
|----------|---|-----------|
| 3.3.2 | Centre of Mass | 38 |
| 3.4 | Source Extraction Using Image Segmentation | 39 |
| 3.4.1 | Methodology | 39 |
| 3.5 | Discrete pulse transform of images and applications | 41 |
| 3.6 | The Discrete Pulse Transform | 43 |
| 3.6.1 | Extraction of Extended sources using the LULU operator and the DPT. | 45 |
| 3.7 | Extracting extended sources by image segmentation method using filtering techniques | 48 |
| 3.7.1 | Results of Otsu thresholding | 49 |
| 3.7.2 | Image Filtering | 49 |
| 3.7.3 | Gaussian Filtering | 50 |
| 3.7.4 | Methodology of how Gaussian filtering works | 50 |
| 3.7.5 | Image Denoising | 50 |
| 3.7.6 | Applying Otsu Thresholding on the Gaussian filtered images. | 51 |
| 3.7.7 | Extracting only extended sources. | 52 |
| 3.8 | Remarks | 53 |
| 3.9 | Summary | 53 |
| 4 | Source Classification using Machine Learning techniques | 54 |
| 4.1 | Introduction to Image processing | 54 |
| 4.2 | Shapelets Theory | 55 |
| 4.3 | Sample selection for machine learning classification | 59 |
| 4.4 | Pipeline for Binary Classification using Machine Learning | 60 |
| 4.4.1 | Feature Extraction using the Shapelet Transform | 61 |
| 4.4.2 | Data Visualization | 63 |
| 4.5 | Application of Machine Learning | 65 |
| 4.5.1 | K-Fold Cross-Validation | 66 |
| 4.5.2 | Hyper-parameter optimization and Evaluation Metric | 67 |
| 4.5.3 | Receiver Operating Characteristics (ROC) | 68 |
| 4.6 | Results and Analysis | 69 |
| 5 | Source Classification using Deep Learning | 73 |
| 5.1 | Introduction | 73 |
| 5.2 | Sample Selection for Deep Neural Network | 74 |

| | | |
|----------|---|------------|
| 5.3 | Preprocessing images and data augmentation | 74 |
| 5.4 | Method 1: Reconstructing model images using shapelet coefficients | 76 |
| 5.4.1 | Sampling from a probability distribution | 76 |
| 5.4.2 | Reconstructing model images through sampling from 256 shapelet space | 76 |
| 5.4.3 | Reconstructing Image with different σ s | 78 |
| 5.4.4 | Summary | 80 |
| 5.5 | Method 2 : Image pre-processing and Augmentation | 80 |
| 5.6 | Method 3: Generating Realistic-Looking Radio Images with Adversarial Neural Networks | 82 |
| 5.6.1 | Generative Adversarial Networks (GANs) building blocks | 83 |
| 5.6.2 | Approach and Model architectures | 85 |
| 5.6.3 | Training DCGANs | 85 |
| 5.6.4 | Details of Dataset for DCGAN Training | 87 |
| 5.6.5 | Generating FRI and FRII radio images using DCGAN | 88 |
| 5.7 | Introduction to Convolutional Neural Network | 90 |
| 5.8 | Convolutional Neural Network architecture | 91 |
| 5.8.1 | Classification using CNN | 92 |
| 5.8.1.1 | Features | 93 |
| 5.8.1.2 | Convolutional Layer | 94 |
| 5.8.1.3 | Max Pooling Layer | 94 |
| 5.8.1.4 | Rectified Linear Units and the sigmoid function | 95 |
| 5.8.1.5 | Fully Connected Layer | 96 |
| 5.8.2 | Network model and training | 96 |
| 5.9 | Results and Analysis | 99 |
| 5.10 | Summary | 100 |
| 6 | Conclusion | 102 |
| A | Introduction to Bayes Theorem | 105 |
| B | The algebraic derivation of PCA and the Maximum Variance Formulation | 107 |
| C | Otsu Thresholding | 110 |
| | Bibliography | 113 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Schematic diagram of an Active Galactic Nucleus | 4 |
| 1.2 | Different features of radio galaxy 3C 47 | 5 |
| 1.3 | An FRI galaxy 3C 449 and an FR II galaxy 3C 98 | 6 |
| 1.4 | Parameters needed to evaluate the Fanaroff-Riley ratio | 8 |
| 1.5 | Fit elliptical gaussians using VSAD package on the mosaic J0000M84 | 13 |
| 1.6 | Contrast between the FIRST data and the FIRST catalogue | 15 |
| 1.7 | Extended sources fitted with elliptical Gaussians | 16 |
| 1.8 | The image-level matching applied on NGC 0547 | 17 |
| 2.1 | Examples of MNIST hand-written digits | 22 |
| 2.2 | k -Nearest neighbours classifier | 24 |
| 2.3 | The Random Forest classifier. | 26 |
| 2.4 | McCulloch-Pitts neuron | 28 |
| 2.5 | The architecture of MLP | 30 |
| 2.6 | Machine learning analysis on astronomical data | 33 |
| 3.1 | Local peak detection in the 2MASX J02581124-5243419 image | 35 |
| 3.2 | Location of centroids of different sources | 39 |
| 3.3 | Deblended segmentation image | 40 |
| 3.4 | Application of LULU on an image | 42 |
| 3.5 | Image with impulse noise and smoothing with LU | 43 |
| 3.6 | Illustration of local maximum and local minimum | 44 |
| 3.7 | Image thresholded with $t = \text{mean}(\text{Original image}) + 3\sigma$ | 46 |
| 3.8 | Image thresholded with $t = \text{median}(\text{image})$ | 47 |
| 3.9 | Image thresholded with $t = \text{median}(\text{image}) + 3\sigma$ | 47 |
| 3.10 | Image thresholded with $t = 1\sigma$ | 47 |
| 3.11 | Image thresholded at the noise level of the image | 48 |
| 3.12 | Application of Otsu thresholding on an image. | 49 |
| 3.13 | The process of filtering | 50 |

| | | |
|------|--|----|
| 3.14 | Image denoising using Gaussian filtering | 51 |
| 3.15 | Otsu thresholding contours on Gaussian Filtered image | 51 |
| 3.16 | Extracting extended sources in an image. | 52 |
| 4.1 | One dimensional basis functions | 55 |
| 4.2 | First few 2-dimensional Cartesian basis functions | 57 |
| 4.3 | Shapelet decomposition of J154712+180410 radio image | 57 |
| 4.4 | Generating J154712+180410 radio source image | 58 |
| 4.5 | Constructing the residual image from the model | 59 |
| 4.6 | The plot of the model and the shapelet coefficients | 59 |
| 4.7 | Illustration of the four different classes of radio sources | 60 |
| 4.8 | Framework for Machine Learning Classification | 61 |
| 4.9 | The average value of the coefficients for point-extended sources | 63 |
| 4.10 | The average value of the coefficients for FRI -FRII sources | 64 |
| 4.11 | Illustration of the first three normalized shapelet coefficients for point and ex- tended sources | 65 |
| 4.12 | Illustration of the first three normalized shapelet coefficients for FRI and FRII sources | 65 |
| 4.13 | The ROC curve and the Area Under Curve for Point-Extended classification | 69 |
| 4.14 | The ROC curve and the Area Under Curve for FRI-FRII classification | 72 |
| 5.1 | Sampling new coefficient from a distribution. | 77 |
| 5.2 | MRC0007-287 image decomposed into shapelet coefficients | 78 |
| 5.3 | MRC0020-253 image decomposed into shapelet coefficients | 79 |
| 5.4 | Reconstructed models of MRC0007-287 using new sampling coefficients with different σ s | 79 |
| 5.5 | Reconstructed model of MRC0020-253 using new sampling coefficients with dif- ferent σ s | 80 |
| 5.6 | Image pre-processing stages with sigma clipping statistics | 81 |
| 5.7 | Data augmentation using flipping and rotation | 81 |
| 5.8 | Illustration of normal convolution | 83 |
| 5.9 | A demonstration of fractionally-strided convolution | 84 |
| 5.10 | The architecture of the DCGAN generator | 84 |
| 5.11 | The architecture of the discriminator. | 86 |
| 5.12 | DCGAN FRI simulated Radio Images | 88 |

| | |
|--|-----|
| 5.13 DCGAN FRII simulated Radio Images | 89 |
| 5.14 An illustration of a regular 3-layers Neural Network | 90 |
| 5.15 Layers in a convolutional neural network | 91 |
| 5.16 Classification of FRI and FRII using CNN | 92 |
| 5.17 Illustration of pixel values for two different images | 93 |
| 5.18 The features allocated for FRI and FRII source | 93 |
| 5.19 A demonstration of convolution | 94 |
| 5.20 Application of Max Pooling on an image | 95 |
| 5.21 Two nonlinear activation functions | 95 |
| 5.22 ConVNet architecture implemented for this study | 97 |
| 5.23 The ROC curve and the AUC value for the ConvNet classifier. | 100 |
| 5.24 Some examples images in the validation samples | 100 |
| C.1 Greyscale image of 6-level and its histogram plot | 110 |
| C.2 Pixel values correspond to background | 111 |
| C.3 Pixel values correspond to foreground | 111 |

List of Tables

| | | |
|-----|---|-----|
| 1.1 | Parameters obtained from VSAD for catalogue 2MASX J02581124-5243419 | 14 |
| 1.2 | Samples Data | 18 |
| 3.1 | The pixel coordinates and the peak values of the detected sources in the 2MASX J02581124-5243419 image. | 35 |
| 3.2 | Properties of the sources in the 2MASX J02581124-5243419 image | 36 |
| 4.1 | Table summarizing the sample selection for the machine learning algorithms . . . | 60 |
| 4.2 | Confusion matrix for both classifications | 68 |
| 4.3 | A summary of the performance results of the various classifiers for point & extended sources | 70 |
| 4.4 | A summary of the performance results of the various classifiers for FRI & FR II sources | 71 |
| 5.1 | Table summarizing the sample selection for deep learning algorithm | 74 |
| 5.2 | Some notations of the probability distribution. | 85 |
| 5.3 | A summary of the ConvNet architecture | 97 |
| 5.4 | A summary of the dataset size used for training the ConvNet model | 98 |
| 5.5 | A summary of the performance results of the ConvNet model for FRI-FR II classification | 99 |
| C.1 | The within class variance for all the possible thresholds | 112 |

ACKNOWLEDGEMENT

Every project big or small is successful largely due to the effort of a number of wonderful people who have always given their valuable advice. I sincerely appreciate the inspiration; support and guidance of all those people who have been instrumental in making this project a success. I feel deeply honored in expressing my sincere thanks to my supervisors, Prof Bruce Bassett and Dr Nadeem Oozeer for making the resources available at the right time and providing valuable insights leading to the successful completion of my project. I am fully indebted to them for their great patience and constructive reviews in the achievement of this research work. Without their guidance, this study would not have been possible.

Beside my supervisors, I would like to thank Prof Illani Loubser for her comments and guiding principles. Additionally, I owe sincere gratefulness to Etienne and Shankar for their insightful encouragements and comments, but also for answering all my unending questions which helped me to widen my research from various perspectives. My sincere thanks also goes to Michelle for understanding me and providing moral support in difficult times.

I would also like to thank the SKA SA, North-West University and NASSP for providing financial support for my MSc. I would also thank the director of NASSP, Kurt, for his constant support and help. Thanks to Griffin Foster, Inger Fabris-Rotelli and Stephan van der Walt for replying to all my emails and for their valuable advice.

I would also like to thank Bruce again for providing me such a great opportunity to join his dynamic group at AIMS and to meet wonderful people. I would like to thank Yabebal, Rene, Pierre-Yves, Emmanuel, Ethan, Alireza, Kimeel, Kayode and Eli for making the cosmological group at AIMS a welcoming and warm atmosphere. I would like to thank Sheean for guiding me in my research and keeping me motivated in hard times.

I express my deepest appreciation to my mom, brothers and sister who have been my constant source of inspiration during the preparation of this project work. Most importantly, I express my profound gratitude to Harry for providing me with unfailing support, continuous encouragement, to be always by my side in both good and bad times and for providing me unending inspiration.

Thanks for all your encouragement!

ABSTRACT

Until now radio galaxies have primarily been classified using the human neural system. The Square Kilometre Array (SKA) will, however, produce a very large amount of science data, extending into the multiple-petabyte range. Therefore there is an urgent need to develop new, automated techniques to maximally exploit the SKA data. Machine Learning (ML) techniques are currently being used in several fields of Astrophysics and in this thesis we comprehensively explore ML as a way to distinguish point and extended sources (P-E) and to classify radio galaxies as belonging to Fanaroff-Riley class I or II (FRI-FRII).

Our first step was to classify radio sources based on their morphology using filtering methods. We used images from the Sydney University Molonglo Sky Survey (SUMSS) and compared the following techniques: (i) the LULU operators and the Discrete Pulse Transform (DPT) algorithms with a low and high pass filtering. The LULU and DPT algorithms have only been successful in classifying extended sources and are computationally expensive. (ii) we then explored other techniques to extract the sources by applying a high pass filter to the radio images. Using Otsu thresholding and Gaussian filtering methods, we have been able to extract not only extended sources but also made gains in computational time.

Our next approach has been to classify P-E and FRI-FRII sources using various ML algorithms. These included the Multi Layer Perceptron (MLP), Random Forest (RF), k -Nearest Neighbours (k NN) and Naive Bayes (NB) which require specific features of the radio images as inputs. We implemented shapelet analysis to decompose the radio images into their corresponding shapelet coefficients which are then fed into the ML algorithms. For P-E discrimination, a neural network was the most effective algorithm, with an accuracy of 89% and area under curve (AUC) value of 93%. For FRI-FRII sources, the RF algorithm proved to be the best with an accuracy of 75% and AUC value of 74%.

The final stage of this thesis has been to apply deep learning to FRI-FRII source classification in the form of a Convolutional Neural Network (CNN). For the first time in radio astronomy we have added a Generative Adversarial Neural (GAN) network to generate realistic looking data to supplement the real data during training. The result from the CNN+GAN algorithm has proved to be better than both the RF algorithm and the CNN alone with standard data augmentation (flipping and rotation), yielding an accuracy of 84% and AUC value of 85%, showing that combining GANs with convolutional networks for radio astronomy is likely to add significant value in the era of the SKA.

Keywords: LULU operators, Discrete Pulse Transform, Otsu thresholding, Multi Layer

Perceptron, Random Forest, K-Nearest Neighbours, Naive Bayes, Shapelet transform, Convolutional Neural Network and Deep Convolutional Generative Adversarial Network.

ONGOING PUBLICATIONS

1. **“No evidence for extensions to the standard cosmological model”** by **Heavens et al. (2017)** in which I am a co-author, has been accepted in the Physical Review Letter Journal and the current version of the paper can be found at [DOI: PhysRevLett.119.101301](https://doi.org/10.1103/PhysRevLett.119.101301).
2. **“Marginal Likelihoods from Monte Carlo Markov Chain”** has been submitted to the Bayesian Analysis Journal and can be found at [arXiv:1704.03472](https://arxiv.org/abs/1704.03472).
3. **“Source classification using Deep Learning”**, a combination of Chapter 4 and 5, is still under preparation and is done jointly with my supervisors: Prof Bruce Bassett & Dr Nadeem Oozeer, and a Post-Doc at AIMS: Etienne Vos.
4. **“Point source detection using Deep Learning”** is still under preparation and involve collaborative work with my supervisors: Prof Bruce Bassett & Dr Nadeem Oozeer, a Resident Researcher at AIMS: Dr Michelle Lochner, a Post-Doc at AIMS: Etienne Vos and a PhD student at Shahid Beheshti University, Alireza Vafaei Sadr.

ABBREVIATIONS

| | |
|--------------|--|
| AGN | Active Galactic Nuclei |
| AUC | Area Under Curve |
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| CNN | Convolutional Neural Network |
| DPT | Discrete Pulse Transform |
| DL | Deep Learning |
| DNN | Deep Neural Network |
| DCGAN | Deep Convolutional Generative Adversarial Neural Network |
| FIRST | Faint Images of the Radio Sky at Twenty centimeters |
| FRI | Fanaroff Riley I |
| FRII | Fanaroff Riley II |
| FPR | False Positive Rate |
| kNN | k-Nearest Neighbours |
| MOST | Molonglo Observatory Synthesis Telescope |
| ML | Machine Learning |
| MLP | Multi-Layer Perceptron |
| NRAO | National Radio Astronomy Observatory |
| NVSS | NRAO VLA Sky Survey |
| NB | Naive Bayes Classifier |
| PCA | Principal Component Analysis |
| ReLU | Rectified Linear Units |
| RF | Random Forest |
| ROC | Receiver Operator Characteristic |
| SKA | Square Kilometre Array |
| SMBH | Super Massive Black-Holes |
| SUMSS | Sydney University Molonglo Sky Survey |
| TPR | True Positive Rate |
| VLA | Very Large Array |
| 3C | Third Cambridge Catalogue |

List of symbols and Physical Constants

| | |
|-----------------|---|
| Hubble Constant | $H_0 = 67.80 \pm 0.77 \text{ km s}^{-1} \text{ Mpc}^{-1}$ |
| Electron Charge | $e = 1.60217662 \times 10^{-19} \text{ C}$ |
| Speed of Light | $c = 2.99792458 \times 10^8 \text{ m s}^{-1}$ |
| Lorentz Factor | γ |
| Magnetic Field | B |
| Frequency | f |

Chapter 1

1 Introduction

Over the last few decades, the techniques we adopt to do science have changed tremendously with the exponential growth of data. Astronomy and astrophysics are also participating in this explosion with the development of increasingly sophisticated facilities, both on the ground and in space. With such a rapid pace of advancement in technology, massive amounts of data are produced that will reach from a range of terabytes to petabytes in the near future. This production rate of data, in terms of variety (complex data), volume (petabytes of data) and velocity (production and transmission rate) is leading astronomy into the era of big data, especially radio astronomy. Hence, there is a need for new and advanced processing solutions, such as Machine Learning (ML) and Deep Learning (DL) algorithms to search for hidden patterns in data far beyond what humans can process. In this study, our main focus is based on the classification of the morphology of radio sources from large astronomical surveys, with a view towards the Square Kilometre Array (SKA).

1.1 A brief history of Radio Astronomy

Karl Guthe Jansky in the 1930s ([Jansky 1933](#)) made the first detection in radio astronomy with the serendipitous discovery of radio emission at a frequency of 20.5 MHz or a wavelength of 14.6 m from the centre of the Milky Way Galaxy. Following the footsteps of [Jansky \(1933\)](#), Grote Reber, a pioneering radio engineer carried out the first systematic survey of the radio universe and observed radio emission from our galaxy, but it was not until after the second World War that radio astronomy became a significant field of its own.

In 1950, a confirmed approach about extragalactic radio emission came into being. This was characterized by Brown and Hazard who observed radio emission emitted by M31 ([Brown & Hazard 1950, 1951](#)) which is the big spiral galaxy in Andromeda by using the data obtained at

the Jodrell Bank Observatory. [Jennison & Das Gupta \(1953\)](#) discovered the first powerful radio galaxy, which was later named as Cygnus A. This radio galaxy showed a “double radio-lobed” structure, with each lobe on opposite sides of an optical galaxy. This remarkably added much development to the flourishing era of radio astronomy, after which more radio sources were discovered.

Over the last three decades, astronomers have conceived and built special telescopes and instruments that helped in the development of various surveys for example the series of Cambridge surveys. One of the most important techniques developed for radio astronomy is radio interferometry. This consists of multiple antennas interconnected together which sample and digitize the incoming radio waves. It uses the principle of superposition to overlay two or more waves, hence producing an interference pattern, from which information about the constituent waves can be extracted.

With the forthcoming modern interferometric arrays, our understanding will drastically improve. The SKA ([Lazio 2009](#)) will consist of various types of arrays that will cover frequency ranges largely covered by existing instruments. Precursors such as the South African MeerKAT ([Booth et al. 2009](#)) and the Australian Square Kilometre Array Pathfinder (ASKAP) ([Johnston et al. 2008](#)) will eventually be merged with the massive SKA project. New instruments and surveys that are planned or entering operations in the near future indicate that radio astronomy will play an important role for the advance of astrophysics. Another aspect that will enable astrophysics to develop in large leaps is the ever growing computing capability.

1.2 Astronomical Sources

Depending on the type of astronomical object, sources present themselves with different sizes and morphologies. The angular resolution, θ , is the minimum angular scale between two point sources which the telescope can successfully distinguish as two separate sources, and is given by Equation 1.1:

$$\theta \approx 1.22 \frac{\lambda}{D} \quad (1.1)$$

where λ is the observing wavelength, θ is the angle, measured in radians and D is the diameter of the telescope ([Lord Rayleigh 1879](#)). When the angular size of a source is smaller than the angular resolution of the telescope, the source appears as a point source. The source is unresolved by the telescope. The response of a telescope to a point source is called the

point spread function (PSF) (Bradt 2004), a two-dimensional distribution of intensity in the telescope's focal plane. Therefore, the appearance of a point source in an image is given by the convolution of the source brightness distribution with the PSF.

On the other hand, extended sources are sources with angular sizes larger than the telescope's angular resolution. Some extended sources have compact spherical shapes (e.g galaxies) and can be irregular (e.g supernova remnants). One of our goals is to develop an automatic classification of radio-source population. Thus the properties of radio sources, their structure and their connection to the unified Active Galactic Nuclei (AGN) model is relevant and we therefore review them in the following sections. It should be noted for the following sections, where not mentioned otherwise, the main sources of references were from Donoso (2010).

1.3 Galaxies

Galaxies are gravitationally bound systems consisting of a billion to a hundred billion stars, an interstellar medium of gas and dust (dark matter). Astronomers used a galaxy morphological classification to divide galaxies into groups based on their visual appearance. Galaxies are divided into two main groups (Hubble 1926):

1. Normal galaxies: Normal spiral, barred spiral, elliptical, peculiar and irregular galaxies.
2. Active galaxies: Seyfert galaxies, Radio galaxies, quasars, starburst and BL Lacertae.

1.3.1 Active Galactic Nuclei

Active Galactic Nuclei (AGN) discovered by Carl Seyfert in 1943 (Seyfert 1943), are among the most interesting natural phenomena in the Universe in relation to their description as objects since they have central regions with peculiar optical spectra and are associated with Super Massive Black Holes (SMBH).

AGNs are compact, intrinsically luminous regions located at the centres of galaxies. An AGN can produce more radiation than the rest of the galaxy at radio wavelength (Donoso 2010). It is believed that all AGNs are powered by accretion onto SMBH (Lynden-Bell 1969). Some AGNs have strong luminosities and dominate most objects at high redshifts. Moreover, AGN often produce highly collimated structures called Jets. Jets consist of fast moving particles that are mostly produced from inner regions of accretion disks in the AGN centres. Jet-driven radio structures can extent for 10 s to 100 s of kpc beyond the host galaxy. Jets radiate in all wavebands but they are more obvious in radio observations.

AGN can be classified into two groups based on the radio properties: radio-quiet AGN and radio-loud AGN. Based on their physical and observational properties, AGN can be further classified into sub-classes: Seyfert galaxies, quasars, blazars and radio galaxies. In this work, we will mainly focus on the morphology of radio galaxies and will not sub-categorize them.

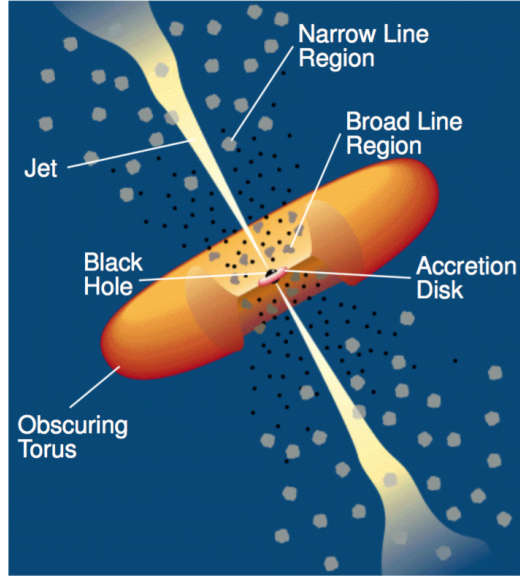


Figure 1.1: Schematic diagram of an inner structure of Active Galactic Nucleus (AGN). The central region of the accretion disk is energetic due to presence of a black hole. In addition, high energy particles are confined to well collimated jets, and are emitted into extragalactic space. Figure courtesy of [Urry & Padovani \(1995\)](#).

1.3.2 Radio Galaxies

The extragalactic radio sky is mostly dominated by radio galaxies at flux densities $S_{1.4\text{GHz}} \gtrsim 1$ mJy. At the centre of a radio galaxy, for instance, accretion of matter onto the SMBH fuel those powerful sources, thus resulting in the production of an accretion disc surrounded by dust. Figure 1.1 illustrates how broad emission lines are formed by clouds of gas moving rapidly, close to the central black hole while narrow emission lines are produced by slow moving gas, further from the accretion disc. Radio galaxies emit radio radiation from nuclear and extended structures which are associated with the synchrotron process, thus providing us with information about how AGN evolve and interact with their environment. Typically, radio galaxies are associated with elliptical galaxies ([Ocana et al. 2008](#)).

1.3.3 Morphology and structure of radio galaxies

At radio wavelengths, radio galaxies show a variety of sizes and morphologies with lobes, radio halos, nuclei, jets and filaments as shown in Figure 1.2. Radio galaxies exhibit steep spectrum radio sources and follow the power law spectra as given by Equation 1.2.

$$S \propto \nu^\alpha \quad (1.2)$$

where α is the spectral index, S is the flux density and ν is the frequency. Compact sources always show flat spectrum with $-1.0 \lesssim \alpha < 0.5$ and extended emissions show steep spectra with $\alpha \geq 0.5$ (Edwards & Tingay 2004).

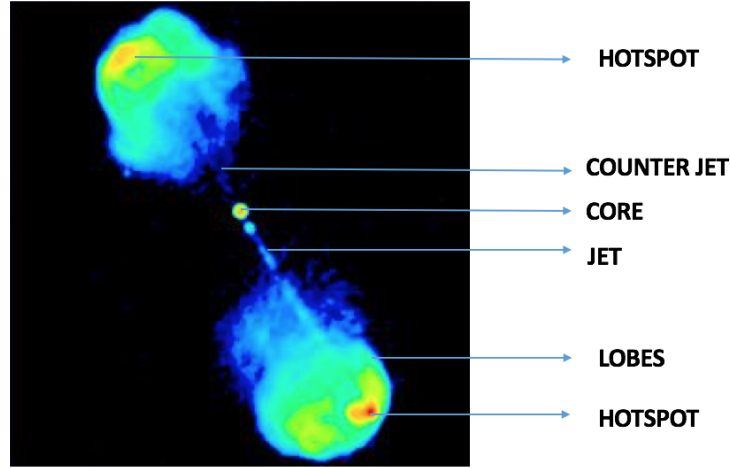


Figure 1.2: A pseudo color image of the FR II radio galaxy 3C 47. The different features: hotspot, jets, lobes and core are labelled.

In many cases, the morphologies of radio galaxies are too complex to be able to distinguish the different observed components. However, the most common features observed are (Donoso 2010):

- **Core:** The core is a compact component having a flat, self-absorbed spectrum and is present in almost 80% of radio galaxies.
- **Jets:** The jets are narrow beams of plasma that transport particles and energy from the central AGN to the extended radio lobes. The jet emission is powered by synchrotron radiation (Schwinger 1949) which is emitted from charged particles that are gyrating at relativistic speeds around magnetic field lines (Shklovsky 1958). Jets can be one-sided or two-sided of the radio galaxy, having a smooth or knotty structure. Radio jets are seen to have steep spectra which are highly collimated close to the core and the magnetic field is parallel to the jet direction. In some cases, jets are observed on both sides where one side is much brighter than the other and this is also called a counter-jet as shown in Figure 1.2.
- **Lobes:** These are large radio-emitting regions ranging from Kpc to Mpc in linear sizes. Two radio lobes are mostly located in a symmetric configuration at opposite sides of the

core, assuming no orientation effect. The angle between the two lobes with respect to the core is known as the opening angle and this angle is around 180° in classical radio galaxies while very small for narrow-angle-tail radio galaxies.

- **Hotspots:** They are mostly located in the outer edges of radio lobes having maximum intensities. Bright radio galaxies can show multiple, one or no hotspots at all. More often, the spectrum is less steep at the hotspots compared to the hosting lobe.

1.3.4 Morphological Classification of Radio sources

Finding a correlation between the morphological type of radio galaxies and their radio luminosity is an important issue in observational astronomy. This approach was initiated in 1974 by [Fanaroff & Riley \(1974\)](#) who found an intriguing correlation between radio luminosity and the radio morphology of the brightest peaks in radio maps of quasars and galaxies.

They noted that the radio galaxies fell into two distinct sub-classes: the Fanaroff-Riley class I (FRI) and the Fanaroff-Riley class II (FRII).

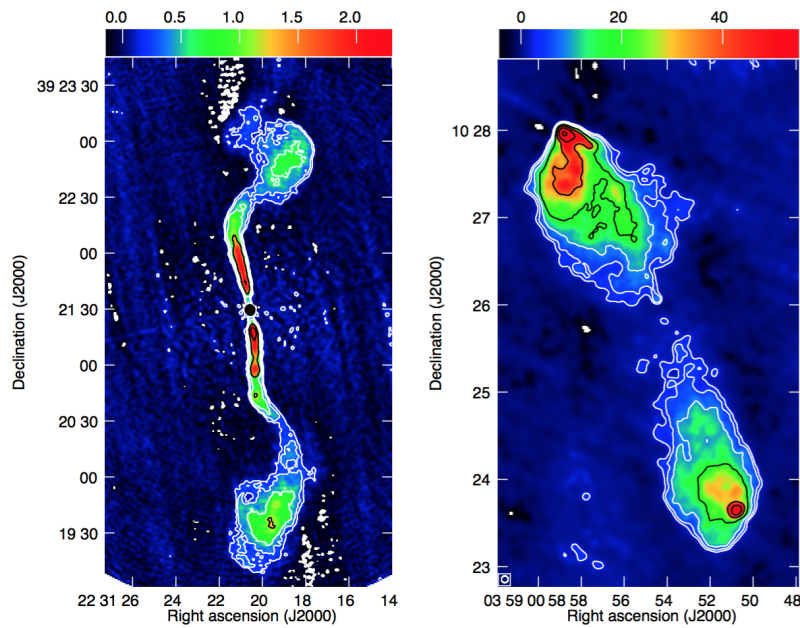


Figure 1.3: The left panel illustrates the FRI galaxy 3C 449 and on the right panel is the FRII galaxy 3C 98. The red indicates the brightest radio emission, for FRIs it is closest to the radio core while for FRIIs it is furthest away from the core, at the termination points of the jets. Figure from [Kharb et al. \(2015\)](#)

1.3.4.1 Fanaroff-Riley Class I (FRI)

FRI are low-power radio galaxies that become fainter towards the outer regions of the lobes. They show high intensity peaks at the central nucleus and the regions of low surface brightness

lie further away. FRI sources are distinguished by the presence of the extended plumes and tails with no distinct termination of the jet. The latter is clearly visible in the radio map of the source 3C 449 as shown in the left panel of Figure 1.3.

Around 80% of FRI objects consist of radio jets (Colina & Perez-Fournon 1990) and an increase in the steepness of the spectra is seen towards the outer regions which indicates that the radiating electrons are comparatively old. In addition, FRI sources are associated with rich clusters filled with hot, X-ray emitting gas and are often hosted by bright, elliptical D/cD galaxies (Zirbel 1996).

1.3.4.2 Fanaroff-Riley Class II (FR II)

FR II are the high-power radio galaxies that end in bright hotspots, which are located at large distances from the core when compared to the total extension of the radio source. With sufficient observational resolution, FR II show narrow, well-collimated jets with clear termination points and jets as shown in the right panel of Figure 1.3. Also, FR II optical hosts are usually giant elliptical galaxies.

One of the differences between FRI and FR II sources is their jets. FRI sources have jets that are wide, knotty and are distorted by the ambient medium showing that they have been decelerated to subsonic speeds (Perucho & Marti 2007). On the contrary, FR II sources have jets that are narrow, smooth, collimated and terminate in bright hotspots at the edges, demonstrating they flow with supersonic speeds (Eilek 2014). Such behaviour arises from two different scenarios.

1. **Environment:** radio galaxies have powerful radio jets that are disrupted and decelerated to subsonic speed after impacting with the ambient medium (Perucho et al. 2014).
2. **Central Engine:** The changes in radio jets are attributed to the different properties or intrinsic mechanisms of the central engine powering FRI and FR II sources. Baum et al. (1995) showed that at fixed absolute magnitude or radio luminosity, far more optical line emissions are seen with FR II galaxies compared to FRI sources. Moreover, it is found that radio and emission line luminosity are correlated in powerful FR II, demonstrating that they are somehow linked by some physical process. The bottom line that can explain the differences between the two class objects are the differences in accretion rate, black hole mass, or black hole spin rate (Donoso 2010).

1.3.4.3 Differences between FRI and FR II

Fanaroff & Riley (1974) found the following major differences between the two classes:

1. This simple classification scheme under which radio galaxies fall into two distinct subclasses was not only based on their radio morphology. The structures of radio galaxies seem to undergo an abrupt transition around total radio luminosity $P_{178\text{MHz}} = 5 \times 10^{25} \text{ WHz}^{-1}$ for Hubble constant, $H_0 = 100 \text{ Kms}^{-1}\text{Mpc}^{-1}$. The category of sources below this critical luminosity was known as FRI and on the other hand, those beyond were FR II type objects.
2. Apart from having a critical luminosity, one crucial aspect which differentiates between an FRI and an FR II radio source is the Fanaroff-Riley ratio (R_{FR}). Using a sample of only 57 resolved radio sources selected from the Third Cambridge Revised (3CR) catalogue, the R_{FR} is defined as the ratio of the separation between the two brightest regions to the total source size, which is characterized by the outermost detected features at the extremity of the source as illustrated in Figure 1.4 and Equation 1.3 where for FRI sources, $R_{FR} < 0.5$ and for FR II sources, $R_{FR} > 0.5$. R_{FR} is defined by

$$R_{FR} = \frac{B}{A} \quad (1.3)$$

Currently, identifying radio sources is done using radio images and optical overlays for follow up multi-wavelength analysis. However, with the huge data sets coming online, automated detection and classification of these objects is crucial. In the following section, we review some of the source detection tools used by astronomers.

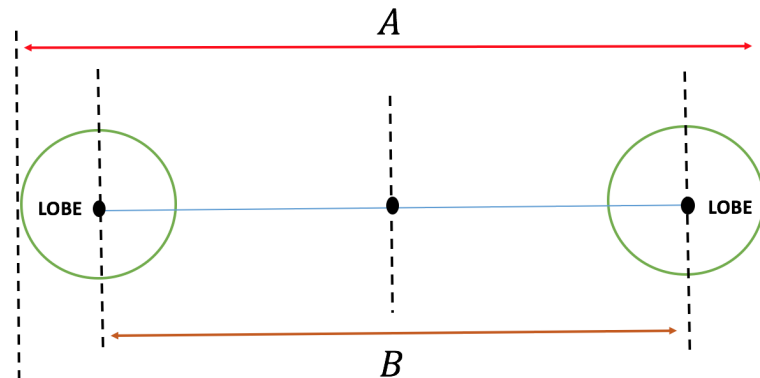


Figure 1.4: A schematic diagram showing the parameters needed to evaluate the Fanaroff-Riley ratio.

1.4 Review of some source detection techniques

In this work, we will later apply some selected methods to detect and extract sources in radio images. In this section, we present some techniques that have been used by other researches.

Source detection techniques are mainly directed towards two main classes, namely basic detection algorithms and multi-scale approaches. Basic detection algorithms are mostly focused on local peak search, thresholding, segmentation, background estimation and filtering while multi-scale approaches are mostly based on the wavelet transform. Barreiro et al. (2003) have applied several filters (e.g Mexican hat wavelet, matched filter (MF) and the scale-adaptive filter) to optimize the detection of objects using a local peak search. Some of the most widely used filtering strategies are focused on wavelet and curvelet transform, deconvolution using regularized linear method, Bayesian methods and wavelet-based deconvolution.

Recently, in astronomical data analysis, Starck & Bobin (2010) have also investigated multi-scale techniques. Their work is mainly focused on wavelet, curvelet and ridgelet transforms. Butler-Yeoman et al. (2016) built an algorithm to detect diffuse sources of any size in an astronomical image. They considered a tree of nested bounding boxes and used an inverted hierarchical Bayesian generative model to obtain the probability of sources existing at given locations and sizes. This model is able to detect nested sources as well. Butler-Yeoman et al. (2016) implemented an algorithm called Oddity which is a detection algorithm that outputs boxes around sources. Oddity is based on a tree-based generative model of an image and finds sources via a tractable Bayesian inversion of this model.

With the advent of innovative techniques in the field of computer vision, various ways are being provided to automatically detect astronomical objects in images. The traditional methods for classification scheme will be introduced and they are based on two main steps: image transformation (see Section 1.4.1) and detection criteria (see Section 1.4.2).

1.4.1 Image Transformation

Image transformation is one of the basic steps one can do to achieve better performance. The most common techniques within image transformation are filtering, deconvolution, application of transform or morphological operations. In astronomical imaging, one of the main objectives of image transformation is to filter the noise, estimating the noise and highlighting objects in some way in the images. Damiani et al. (1997) and Makovoz & Marleau (2006) implemented the median filter to estimate the background noise and to minimize the effect of bright point

source light. In addition, the median filtering was applied by Yang et al. (2008), Perret et al. (2008) and Lang et al. (2010) to filter noise and as a smoothing algorithm on the images.

Another step in astronomical object detection is background estimation. In the optical, there are packages such as DAOPHOT (Stetson 1987) and SExtractor (Bertin & Arnouts 1996) that estimate the local background. This background estimation step is also called σ -clipping which was applied by Vikhlinin et al. (1995), Lazzati et al. (1999) and Perret et al. (2008). Another common image transformation algorithm is to use a Gaussian profile and convolve it with the image. Damiani et al. (1997) applied a Gaussian filter in their multi-scale analysis to smooth spatial variations of the background. Also, this convolution is applied to optical images by Slezak et al. (1999) to enhance faint sources. These techniques can be applied to our data to filter the noise and smooth the radio images.

1.4.2 Detection Criterion

We looked at different detection techniques that will be used for source extraction in this thesis. Once an image transformation is applied on an image, the latter is further used to extract sources and separate them from the background. Therefore, a detection method needs to be implemented. There are two main strategies of detection, namely thresholding and local peak search.

1.4.2.1 Thresholding

When thresholding is applied on an image, a certain cut-off is assigned where connected pixels above that value belong to an object. Thresholding is another way to perform image segmentation where pixel values that are below that threshold are given a value of zero while those above the threshold are assigned a value of 1 as illustrated in a more formal way in Equation 1.4.

$$I_{thresh}(i, j) = \begin{cases} 1 & \text{if } I(i, j) > thresh \\ 0 & \text{otherwise,} \end{cases} \quad (1.4)$$

where the binarized image intensity is represented by $I_{thresh}(i, j)$ and the original image intensity is $I(i, j)$ along the i^{th} row and j^{th} column and $thresh$ is simply the value of threshold applied on the image.

In astronomical fields, thresholding is utilized to detect connected pixels which are considered as sources and those below that threshold are considered as background. However,

deciding an appropriate threshold is a difficult task when taking into account the variation of noise, background or edges of objects in an image. Choosing the threshold is an important step as it may result in some true sources being overlooked (also known as false negatives) or some spurious objects to be considered as real sources (also known as false positives). Irwin (1985) and Freeman et al. (2002) computed the threshold based on the sky estimation while Strack et al. (1998) and Lang et al. (2010) set the threshold to be a multiple of the noise in the image. However, in astronomical image detection, due to variation in background, local or adaptive thresholding are implemented for different regions in the image where a sliding window can be used. An automated thresholding strategy was applied by Yang et al. (2008) known as the Otsu method (Otsu 1979) that utilized a minimized intra-class variance to obtain a good threshold. This method is further discussed in Appendix C.

1.4.2.2 Local Peak Search

The local peak search strategy searches for pixels that are a local maximum in the neighbouring pixels. Often, this step is carried out after the thresholding method has been implemented on the image to avoid unnecessarily analyzing all pixels. The main objective of local peak search is to output a list of candidates, with their associated locations as well as their photometry information. The method of finding the local maxima is mostly used to detect stars and point sources, but is not well suited to detect extended sources and galaxies. This method is illustrated in mathematical way as follows:

$$I_{LPS}(i, j) = \begin{cases} 1 & I(i, j) \geq I(k, l) \\ 0 & otherwise \end{cases} \quad (1.5)$$

where $I(i, j)$ is the pixel intensities in the i^{th} row and j^{th} column, $I(k, l)$ represents the intensity of a neighbour pixel. Herzog & Illingworth (1977) and Newell & O'Neil (1977) implemented this method in the late 70s where they computed the threshold based on the sky level and then considered a maximum pixel as a peak whose intensity is greater than or equal to their eight neighbouring pixels. Hence, the connected pixels are centered on a peak thus considered as a single object. In addition, they have applied Data Over Gradient (DOG) test to deblend sources (Herzog & Illingworth 1977, Newell & O'Neil 1977).

Although most of the classical approaches are focused towards thresholding and local peak search, other techniques have also been applied to detect and extract astronomical sources. During the last few years, these strategies have been developed and are more oriented on tech-

niques from the deep learning, machine learning and computer vision fields. With the advent of new precursors for instance the MeerKAT and ASKAP (Norris et al. 2011), the radio sky is expected to be surveyed at high speed with unprecedented sensitivity, generating high data volumes. It will not be possible to handle this amount of data with manual studies or using classical approaches, and therefore automatic data processing is fundamental.

1.5 Surveys

An astronomical survey is a set of many images or spectra of celestial objects that share common features or types. This enables astronomers to model a catalogue of celestial objects. Then, statistical analyses can be performed on existing surveys that usually lead to new discoveries.

1.5.1 Radio Surveys

Surveys in radio frequencies give astronomers a better understanding of the intensity and distribution of radio sources in the sky. We can classify radio surveys as imaging and discrete source surveys. The Faint Images of the Radio Sky at Twenty centimeters (FIRST), National Radio Astronomy Observatory (NRAO) Very Large Array (VLA) Sky Survey (NVSS) and Westerbork Northern Sky Survey (WENSS) are among the most recent surveys that utilize interferometers. In this project, three surveys were used, namely the Sydney University Molonglo Sky Survey (SUMSS), FIRST and NVSS.

1.5.2 SUMSS

SUMSS was carried out at 843 MHz with the Molonglo Observatory Synthesis Telescope (MOST) that consists of ~ 590 mosaic images of size $(4.3^\circ \times 4.3^\circ)$ with $45'' \times 45'' \cos \delta$ resolution and a source catalogue that covers 8100 deg² of the southern sky. More detailed information about the catalogue is given in Mauch et al. (2003) and a version of the catalogue is available at Astrophysics Research Group: The SUMSS Catalogue¹. The source catalogue is constructed using Astronomical Image Processing System (AIPS) task VSAD (VLA Search And Destroy is a special Gaussian-fitting program) which locates sources and fits elliptical Gaussians in 271 mosaic images to a limiting peak brightness of 6 mJy beam⁻¹ at $\delta \leq -50^\circ$ and 10 mJy beam⁻¹ at $\delta \geq -50^\circ$. Most of the sources in the SUMSS are well fitted by elliptical Gaussians where the VSAD package returned different parameters of each fitted Gaussian, that is, the J2000 right ascension (RA) α , declination (Dec) δ , peak brightness (mJy beam⁻¹), Full width at half maxi-

¹www.astrop.physics.usyd.edu.au/sumsscat/

mum (FWHM) fitted source major (θ_M) & minor (θ_m) axis and the fitted position angle of the major axis. Additionally, the VSAD package constructs a residual image by subtracting each fitted Gaussian from the original image.

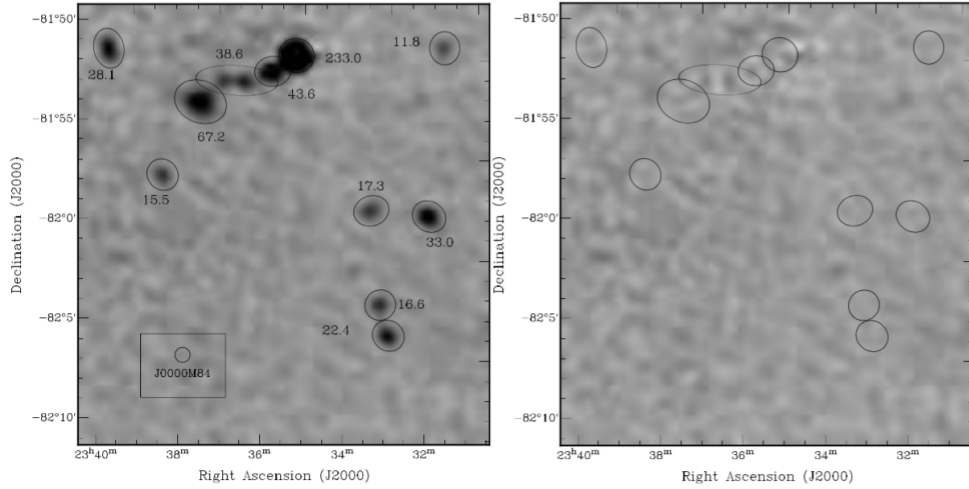


Figure 1.5: On the left, the VSAD package is employed to fit elliptical Gaussians on a small section of the mosaic J0000M84 and for each source, the total flux density (in mJy) is printed beside it. The beam is drawn as a small circle on the bottom left of the image. After subtracting the fitted Gaussians from the input image, the residual image is shown on the right. (Mauch et al. 2003)

Figure 1.5 illustrates a small section of the SUMSS mosaics J0000M84² fitted with elliptical Gaussians using VSAD. It is observed that on the original mosaic, most sources are well fitted with Gaussians. However, some artefacts which are close to stronger sources are also fitted. From Figure 1.5, it is also noticed that for close pairs of sources, VSAD can be unreliable. For example, the 38.6 mJy extended sources are actually two distinct sources, however they are wrongly fitted with a single Gaussian with major axis greater than the true separation of the sources.

To classify the sources as either point or extended, the beam calibration uncertainty is estimated to be $\epsilon_\theta = 3\%$ in both major and minor axes of the MOST beam shape. To deduce if a source is resolved along either axis, the beam $+ 2.33\sigma(\theta_{M,m})$ is compared with the length of the major and minor fitted axes. To identify between point and extended sources in the catalogues, Mauch et al. (2003) have given a deconvolve size to represent extended sources and for point sources, they did not provide source sizes as illustrated in *Columns (7) & (8)* in Table 1.1.

Table 1.1 describes the components that make up the radio morphology of one of SUMSS catalogue: 2MASX J02581124-5243419 and a short description of the columns of the catalogue

²The naming scheme for SUMSS mosaics is *JhhmmMdd* where *J* signifies J2000 coordinates, *hhmm* is the RA in hours and minutes of the mosaic centre, *M* signifies southern declination and *dd* is the declination of the mosaic centre in degrees.

is given as follows.

Columns (1) & (2): The right ascension (*RA*) and the declination (*Dec*) of the source in J2000 coordinates.

Column (3): Peak brightness at 843MHz (in mJy beam⁻¹).

Column (4): Total flux density at 843MHz.

Columns (5) & (6): Fitted major and minor axes.

Columns (7) & (8): Fitted major and minor axes after deconvolution.

Moreover, a decision tree algorithm has been implemented to classify image artefacts which identifies and rejects correctly spurious sources. It is found that 7000 sources from this catalogue overlap similarly with NVSS at 1.4 GHz.

Table 1.1: The different parameters obtained from VSAD for catalogue 2MASX J02581124-5243419.

| RAJ2000 ("h:m:s") | DecJ2000 ("d:m:s") | Sp (mJy) | St (mJy) | MajAxis (arcsec) | MinAxis (arcsec) | dMajAxis (arcsec) | dMinAxis (arcsec) |
|----------------------|-----------------------|-------------|-------------|---------------------|---------------------|----------------------|----------------------|
| 02 58 01.41 | -52 45 01.8 | 58.7 | 143.5 | 95.2 | 66.2 | 83.6 | 34.3 |
| 02 58 15.50 | -52 40 58.7 | 53.7 | 56.0 | 57.1 | 49.5 | 0.0 | 0.0 |
| 02 58 06.70 | -52 40 30.4 | 16.0 | 83.0 | 121.9 | 109.4 | 113.3 | 93.3 |
| 02 58 31.69 | -52 42 36.6 | 8.4 | 9.6 | 61.9 | 49.0 | 0.0 | 0.0 |
| 02 58 09.61 | -52 47 18.5 | 24.5 | 122.7 | 127.2 | 102.7 | 118.3 | 86.4 |
| 02 58 48.45 | -52 47 38.3 | 48.0 | 61.6 | 59.9 | 55.3 | 0.0 | 0.0 |
| 02 58 55.31 | -52 47 08.5 | 73.8 | 83.9 | 58.6 | 49.9 | 0.0 | 0.0 |
| 02 58 40.26 | -52 35 58.1 | 49.9 | 52.4 | 57.1 | 49.3 | 0.0 | 0.0 |
| 02 56 45.37 | -52 42 20.8 | 13.8 | 28.1 | 98.2 | 54.7 | 83.8 | 0.0 |
| 02 59 12.84 | -52 33 13.2 | 29.3 | 30.6 | 57.1 | 51.5 | 0.0 | 0.0 |

1.5.3 The FIRST Survey

The survey covers 10,000 deg² to a sensitivity of ~ 1 mJy with an angular resolution of $\sim 5''$ using the B configuration of the VLA at a frequency of 1.4 GHz (Becker et al. 1995). Sources in the FIRST survey are generated by an AIPS-based source extraction system.

It contains a source extraction program named HAPPY³ which searched pixels in an image that exceeded a threshold value. For each contiguous sample of threshold-exceeding pixels, a minimum-size rectangle is defined which is further padded by a border 3 pixels wide. Then, local maxima are searched in each island as an initial estimated parameters for the Gaussian

³http://sundog.stsci.edu/first/catalog_paper/node3.html

fitting algorithm. Afterwards, the individual islands are analyzed and the fitting algorithms are passed through several criteria. Finally, the program HAPPY (White et al. 1997) gave as output a list of elliptical Gaussian components with the following parameters: the right ascension and declination, peak and integrated flux densities, major and minor axes, and the position angle of the major axis measured east from north (White et al. 1997). Figure 1.6 illustrates an example of complex sources where the right panel shows the Gaussian representation of sources in the FIRST catalogue and the left panel is the image of the FIRST survey. It is observed that the process of the Gaussian deconvolution captured effectively the morphology of these complex sources.

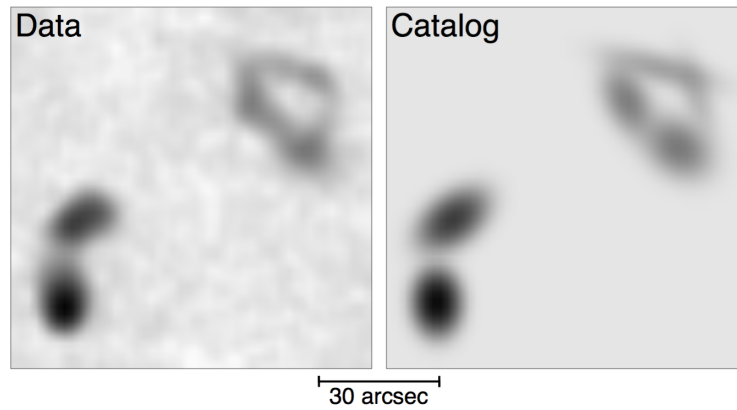


Figure 1.6: Two complex sources utilized to show a contrast between the FIRST data (left panel) and the FIRST catalogue fitted with elliptical Gaussian (right panel). The field is centered at RA = $10^{\text{h}}50^{\text{m}}08.5^{\text{s}}$ and Dec = $+30^{\circ}40'15''$. A bent-double morphology source in the southeast and a peculiar ringlike morphology source are captured in the FIRST Catalogue. Figure from White et al. (1997)

1.5.4 The NVSS Survey

The NVSS covers the sky at declinations north of $\delta = -40^{\circ}$ (J2000) at 1.4 GHz. For this survey, the compact D and DnC configurations of the VLA have been utilized.

The NVSS survey consists of a sample of 2326 ($4^{\circ} \times 4^{\circ}$) continuum cubes made from three planes that contain Stokes I, Q and U images in addition to a catalogue of sources consisting of about 2×10^6 discrete sources stronger than an intensity $S \approx 2.5 \text{ mJy}$. For Stokes I, the fluctuations of their rms brightness are about $\sigma \approx 0.45 \text{ mJy beam}^{-1}$ while for Stokes Q and U, the rms is $\sigma \approx 0.29 \text{ mJy beam}^{-1}$. For the $N \approx 4 \times 10^5$ sources whose intensities are stronger than 15 mJy, the rms uncertainties in the right ascension and declination vary from $\lesssim 1''$ to $7''$. More detailed information of the NVSS survey is found in Condon et al. (1998). Figure 1.7 shows a section of one image from the NVSS survey that contains an extended triple source and various smaller sources. On the top-right panel, a model is constructed from a small amount of

elliptical Gaussians and the bottom panel illustrates the residual image, that is the subtraction between the image and the model.

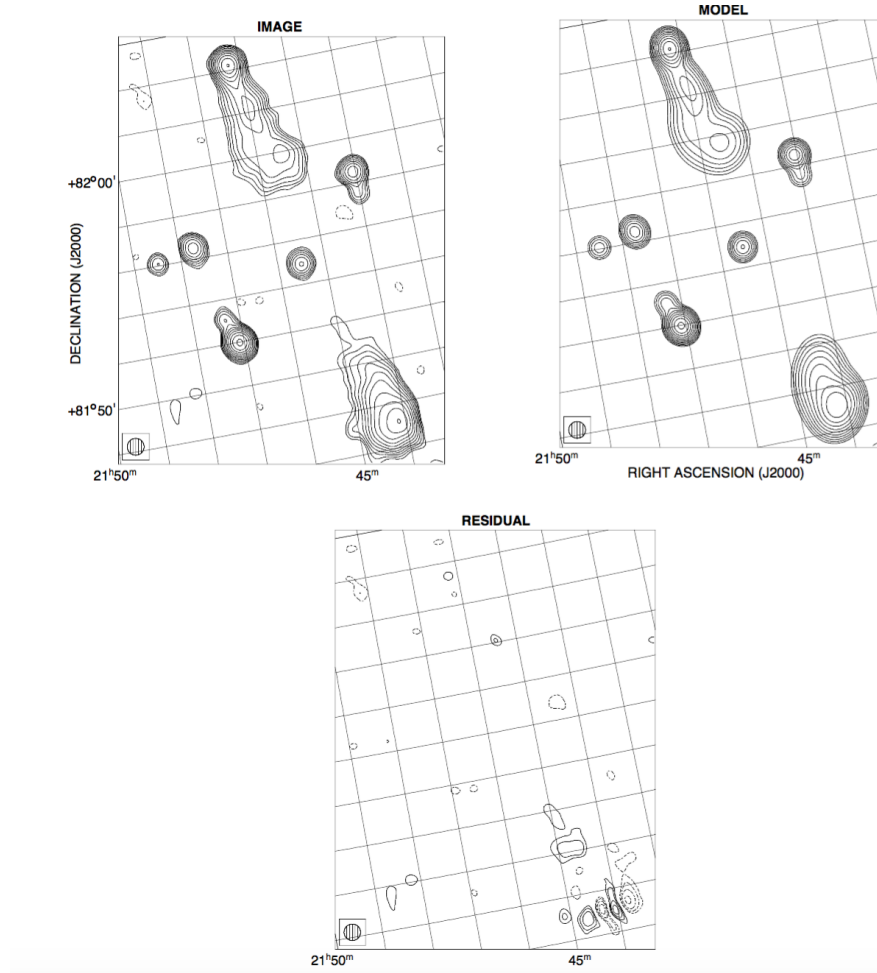


Figure 1.7: The top left panel shows the extended sources in one of the NVSS images. And the extended sources are approximated by elliptical Gaussians as shown in top-right panel. The sources are fitted with contours of $\pm 1, \pm 2^{\frac{1}{2}}, \pm 2^1, \pm 2^{\frac{3}{2}}, \dots \text{mJy beam}^{-1}$. Figure from [Condon et al. \(1998\)](#)

1.6 Sample Selection

This section presents a brief description of the datasets implemented in our work. For point and extended analysis, some filtering algorithms and different ways of source extraction are applied on a sample of images taken from the SUMSS survey. [Van Velzen et al. \(2015\)](#) catalogue was utilized to perform point and extended classification using some machine learning algorithms. Moreover, to perform classification of FRI and FRII classes using machine learning and deep learning concepts, we restricted the sample from the FRICAT ([Capetti et al. 2016](#)) and FRIICAT ([Capetti et al. 2017](#)). These came from the FIRST ([Becker et al. 1995](#)) and NVSS ([Condon et al. 1998](#)) surveys since the radio galaxies are well-resolved and the sources are already classified.

1.6.1 Point and Extended Datasets

The [Van Velzen et al. \(2015\)](#) catalogue contains a sample of 575 radio-emitting galaxies which have a flux greater than 213 mJy at 1.4 GHz. They employed a catalogue-level matching that made use of a friend-of-friend algorithm. They utilized the fitted Gaussians from the NVSS and SUMSS catalogue to match the optical counter part. They used as a criterion a linking length between the fitted Gaussians as given in Equation 1.6.

$$\text{Linking Length} = \max(N_{lim} \times FWHM_i, d_{lim}) \quad (1.6)$$

This allows the entire connected structure of the radio sources consisting of multiple Gaussians to be recovered. After the catalogue level matching, 1273 sources are left. Secondly, false matches were removed from the 1273 sources by implementing an image level rejection which used the stored information within the pixels of the image as shown in Figure 1.8. Cut-outs were made from the NVSS and SUMSS survey where contours were drawn, thus finding pixels within the lowest contours. Pixels whose radii are within $\max(FWHM_i, 30'')$ were then flagged and the centroids of the galaxies were found. Manual inspection and classification was performed where 575 radio galaxies were left. According to their morphology, the galaxies were classified as point sources (97), star-forming galaxies (52), jets and lobes (407) and unknown (19).

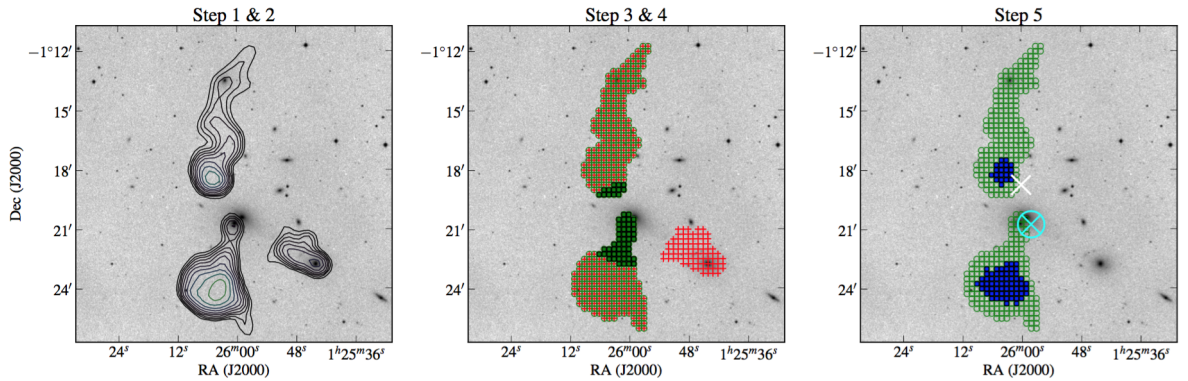


Figure 1.8: The image-level matching applied on NGC 0547. Contours are drawn in step 1 & 2. In step 3, red marks are pixels found above the value of the outer contour. In step 4, the green squares present pixels that are linked to the optical centre. Finally, in step 5, the green circles represent pixels that are connected to a group of pixels obtained in the previous steps. The white cross is the geometric centre of the source and the cyan circled cross is the flux-weighted centre. Figure from [Van Velzen et al. \(2015\)](#).

1.6.2 FRI and FRII Datasets

The FRICAT/FRIICAT catalogue (Capetti et al. 2016, 2017) is used as a subsample of radio sources. Capetti et al. (2016) and Capetti et al. (2017) obtained this catalogue by combining observations from the NVSS, FIRST, and Sloan Digital Sky Surveys (SDSS). They focused on radio sources with an upper redshift limit $z < 0.15$ and applied a morphological classification where they preserved sources only with edge brightened morphology. A further constraint they added is that at least one emission peak of the source should lie at a distance of 30 Kpc from the central host galaxy ensuring the selected sources are well resolved with 5'' resolution of the FIRST samples. FRI and FRII classification was performed individually by the three authors and a source is added to the catalogue if the classification is at least agreed by two authors. The FRICAT catalogue consists of 219 FRI radio galaxies while FRIICAT has 122 FRII radio galaxies.

In our work, we have used a subset of samples of FRI and FRII from the FIRST, NVSS and FRICAT/FRIICAT catalogues. Combining all the sources from these surveys, our sample data consists of 171 FRI sources and 646 FRII sources. The samples of data that will be used for this project is given in Table 1.2.

Table 1.2: Samples of data gathered from various surveys for P-E and FRI-FRII classification.

| Types of Sources | Number of images |
|------------------|------------------|
| FRI | 171 |
| FRII | 646 |
| Point | 78 |
| Extended | 405 |

1.7 Summary

The theoretical background necessary for later chapters is provided in terms of an introduction to radio astronomy and a brief discussion about different types of radio galaxies. A review is provided about some approaches that are employed by different scientists for source detection and source extraction in astronomical images. In addition, an overview of all surveys used for this project is briefly discussed.

1.8 Objectives

As part of the thesis, the main goal is to develop an automatic algorithm to classify radio galaxies, particularly Point-Extended and FRI-FRII galaxies. The general objectives can be divided into two specific parts regarding the different stages of the project:

1. **Source detection in astronomical images.** The evaluation includes a review of some existing methods implemented over the last few years to detect and extract sources from astronomical images, particularly extended sources. We also present the first application of LULU operators (that stands for L (lower) and U (upper)) and the Discrete Pulse Transform (DPT) as a detection algorithm in radio astronomy. In addition, the first application of Otsu thresholding and some filtering methods are presented as an evaluation for the detection and extraction of astronomical sources.
2. **The development of an automated algorithm for source classification.** Our main aim is the development of various machine learning techniques for the classification of sources between FRI-FRII radio galaxies and to distinguish between Point and Extended sources. We have also extended this work to a Deep Learning framework for classification of sources.

1.9 Overview of the thesis

The project is structured and briefly described as follows:

Chapter (1) – Introduction to Radio Astronomy: This chapter introduced the reader to the concepts of radio astronomy and astronomical objects (galaxies and active galactic nuclei) and a clear distinction between FRI-FRII galaxies. A brief review of some source detection techniques is also presented. The surveys and the datasets used in this work, is also described.

Chapter (2) – Introduction to Machine Learning: An overview of the concepts of machine learning is provided. A review on the application of ML used in the astronomy community is also explained.

Chapter (3) – Astronomical source detection using Filter-based methods: The first application of the LULU operators in radio astronomy is presented. We demonstrate the use of various filtering methods for source detection and extraction.

Chapter (4) – Source classification using machine learning techniques: We use the shapelet transform as a feature extraction approach. Using the shapelet coefficients (features) as inputs for the ML algorithms, we provide a classification framework for Point-Extended and FRI-FRII radio galaxies.

Chapter (5) – Source classification using Deep Learning: We provide three approaches for data augmentation in radio astronomy i) first application of shapelet coefficients to reconstruct synthetic images ii) the standard augmentation techniques using rotation, flipping and some transformations iii) the application of Generative Adversarial Networks (GANs) to generate fake images of FRI-FRII sources. Most importantly, we show the classification of FRI-FRII radio galaxies using Convolutional Neural Networks (CNNs).

Chapter (6) – Conclusions: We summarize the concluding remarks drawn from this research work. We also discuss some suggestions for future works.

Chapter 2

We are drowning in information and starving for knowledge. — John Naisbitt

2 Introduction to Machine Learning

With the advent of new instruments and collection of large amount of data (Big Data), Machine Learning has appeared as a favoured tool for astronomers. [Manyika et al. \(2011\)](#) defined Big Data as “datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze”. However, in this definition, there is a time-variant feature. Today we might consider a certain amount of data as Big Data but tomorrow it could be called ‘normal’. Big data is defined by [Laney \(2001\)](#) as “the data growth challenge as three-dimensional, that is, concerning an increase in volume, velocity, and variety”.

With this deluge of data, we therefore need an automated way to perform data analysis. We can define machine learning as a set of methods that can automatically detect patterns in data and then use the discovered patterns to predict future data or to perform other kinds of decision-making under some uncertainties. Nowadays, machine learning has not only become a dominant field in computer science but it has an ever greater role in our everyday life. For example, machine learning is used to empower the robust email spam filters, speech and visual object recognition and many other domains such as genomics, challenging chess players and the efficient autonomous driving cars.

Through the use of computer algorithms, machine learning is concerned with automated detection of regularities in data to take steps such as the classification of data into different categories. Figure [2.1](#) illustrates an example of a typical machine learning problem: the recognition of handwritten digits, Mixed National Institute of Standards and Technology (*MNIST*). It shows the scanned digits that have been normalized. The actual label (human identified label) is shown in green color. Each digit is an 8×8 grid and therefore can be represented with a 64-element vector \mathbf{x} . The aim is to develop a machine learning algorithm that will take the

vector \mathbf{x} as input and will yield as output the identity of \mathbf{x} as the digit 0, . . . , 9. However, the large variability in people's handwriting has made the problem more difficult to solve. An attempt to tackle this problem is to use handcrafted rules to identify the digits based on the structures and shapes.

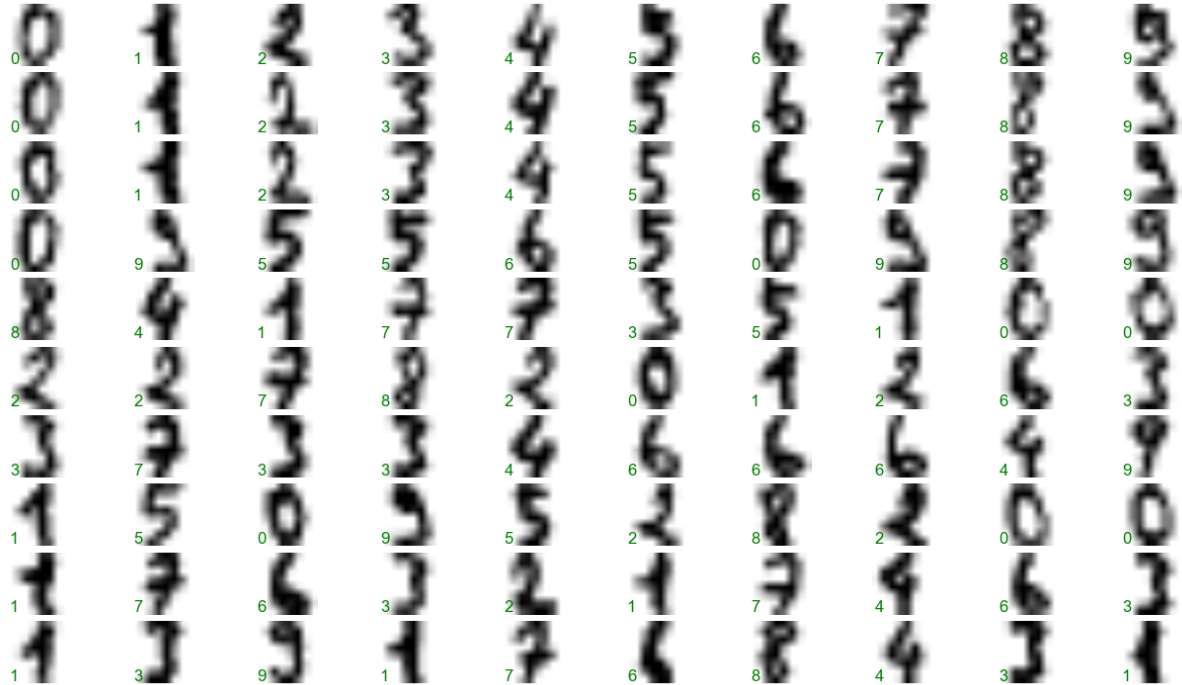


Figure 2.1: Examples of MNIST hand-written digits and the small green characters show the actual label of the digits. Data taken from [LeCun et al. \(1998\)](#).

Therefore, a machine learning approach can be adopted to obtain far better results. A set of K digits $\{\mathbf{x}_1, \dots, \mathbf{x}_K\}$ also known as the *training set* can be used to adjust the parameters of an adaptive system. By inspecting the digits individually, the label of each digit, also referred to as the *target value* t for each digit \mathbf{x} is known in the training set in advance. The output result can be represented by a function $y(\mathbf{x})$, where \mathbf{x} is an input that accepts a new digit image and the machine learning algorithm will output vector \mathbf{y} which is encoded in a similar technique as the target vectors. The specific form of $y(\mathbf{x})$ is achieved in the learning phase (also known as the training phase). After the training phase, the model can be further used to identify digits in the previously unseen *test set*. The ability to correctly classify new examples that differ from those in the *training set* is known as *generalisation*.

2.1 Style of learning

Broadly speaking, machine learning applications can be classified into three broad learning categories: supervised, unsupervised and reinforcement learning. In supervised learning, the

target is to focus on accurate predictions while in unsupervised learning the goal is to create compact description of the data. In both instances, one is focused on methods that perform well with respect to previously unseen data. In reinforcement learning, the aim is to develop a system (agent) that improves its performance based on interactions with the environment. In this work, we will not discuss about reinforcement learning.

2.1.1 Supervised Learning

In supervised learning, a model is learnt from labelled training data that empowers the prediction about unseen or future data. Here, the desired output signals (labels) are already known. We know the right answer beforehand when we train the model.

Supervised learning is based on the mapping from input \mathbf{x} to output \mathbf{y} , given a labelled set of input-output pairs, $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ and N is the number of training examples. The training input \mathbf{x}_i is a \mathbf{d} -dimensional vector of numbers (also known as features, covariates or attributes), that can represent, for example, the height and weight of a person. \mathbf{x}_i can be arbitrary, for example an image, a sentence, an email message, a time series, a molecular shape or a graph. The output \mathbf{y}_i , also known as response variable, is a categorical or nominal variable for a classification problem or is real-valued for a regression problem (Bishop 2000).

2.1.2 Unsupervised learning

Unsupervised learning is also known as knowledge discovery as the main goal is to discover “interesting patterns” in the data. In unsupervised learning, it comprises of unlabelled data or data of unknown structure. This will allow us to explore the structure of the data to extract meaningful information without the guidance of a known outcome variable or reward function. Clustering and dimensionality reduction are forms of unsupervised learning (Bishop 2000).

2.2 Machine Learning Algorithms

For this project, we are concerned with the problem of binary classification, as will be explained in Chapter 4. There are various existing learning algorithms which addresses this problem. In the next section, we provide an introduction to the concepts of a selection of popular machine learning classification algorithms. An intuitive appreciation of the differences between the supervised learning algorithms, their strengths and weaknesses will be given. In the next section, we discuss the random forest, the naive Bayes classifier, the k-nearest neighbours algorithm and the Multi-Layer Perceptron classifier, as these were the algorithms selected for the purpose

of this project. In later chapters, these algorithms will be implemented and their performance will be compared. Where not mentioned otherwise, the main sources of information for the various machine learning algorithms and dimensionality reduction are Bishop (2000), Mitchell (1997) and Gallagher (1999).

2.2.1 k Nearest Neighbours

k Nearest Neighbours (k NN) is a supervised machine learning algorithm which is a non-parametric and instance-based technique utilized for classification. k NN classifies a new unclassified test point x by taking the average class or the majority class vote among the k training points that are nearest to the test point. The k training points are known as the k nearest neighbours (Hastie et al. 2009). k NN calculates the distance between the training sets and the test set maintaining the list of examples of the k nearest training set.

A visualization of the classification of a test point (yellow) into either red or green by a 10-nearest neighbours ($k = 10$) is illustrated in Figure 2.2. In this case, the training sets consist of two-dimensional data points which are either “red” or “green”. In this example, the yellow data point will be classified as red since the latter is nearest to the seven nearest neighbours which belong to the “red” class. Therefore, a majority class vote is assigned to the “red” color.

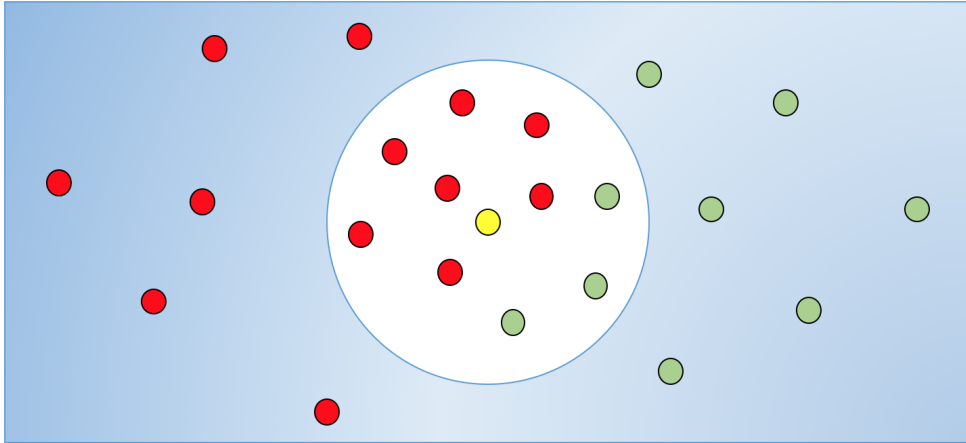


Figure 2.2: Example of k nearest neighbours classifier. The test point (yellow) is classified by a 10-nearest neighbours ($k = 10$) into either red or green. The test point is classified as red as the majority of the 10 neighbours are red and are nearest to the test point.

The operation of k NN is described in Algorithm 1. The k nearest neighbours of a test point are determined by selecting the k training points that are nearest in distance to the test point in feature space. Then, the Euclidean distance D between test point x_0 and a training instance x_j is given by

$$\text{Minkowski} \quad D(\mathbf{x}_0, \mathbf{x}_j) = \sqrt{\sum_{i=1}^d (\mathbf{x}_{0,i} - \mathbf{x}_{j,i})^2} \quad (2.1)$$

$$\text{Manhattan} \quad D(\mathbf{x}_0, \mathbf{x}_j) = \sum_{i=1}^d |\mathbf{x}_{0,i} - \mathbf{x}_{j,i}| \quad (2.2)$$

$$\text{Chebychev} \quad D(\mathbf{x}_0, \mathbf{x}_j) = \max_{i=1}^d |\mathbf{x}_{0,i} - \mathbf{x}_{j,i}| \quad (2.3)$$

where d is the dimension of the input feature space. k NN is a robust algorithm to noisy training set and it performs effectively when the data set is sufficiently large. However, one disadvantage of k NN is that all the features of instances are needed when computing the distance between data points. If a small portion of the data set consists of discriminatory information and the larger portion is irrelevant features, the distance between the instances will be more influenced by the irrelevant features and this problem is known as the *curse of dimensionality*. k NN is sensitive to this problem which can be overcome by weighting each feature differently when computing the distances between instances. Another problem with k NN is efficient memory indexing. For each new classification, significant computation is required as the algorithm slows all processing until a new classification is received. Various techniques such as the kd -tree (Bentley 1975, Friedman et al. 1977) have been developed for more efficient memory indexing of the training data sets.

Algorithm 1 Classification with k Nearest Neighbours (k NN)

Given: Training set $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ with labels $\{y_1, \dots, y_N\}$.

Given: A distance measure $\mathcal{D} : \mathcal{X} \rightarrow \mathbb{R}$.

Given: An integer $0 < K \leq N$.

Given: Test example $\mathbf{x}_0 \in \mathcal{X}$.

Output : Predicted label $\hat{y}_0^{(kNN)}$

1. Let i_1, \dots, i_K be the indices of the k NN of \mathbf{x}_0 in \mathbf{x} with respect to \mathcal{D} , that is,

$$\mathcal{D}(\mathbf{x}_0, \mathbf{x}_{i_1}) \leq \dots \leq \mathcal{D}(\mathbf{x}_0, \mathbf{x}_{i_K})$$

and

$$\mathcal{D}(\mathbf{x}_0, \mathbf{x}_{i_k}) \leq \mathcal{D}(\mathbf{x}_0, \mathbf{x}_i) \quad \text{for all } i \notin \{i_1, \dots, i_K\}$$

2. For each $y \in \mathcal{Y}$, let $\mathbf{x}'_y = \{i_k \mid y_{i_k} = y, 1 \leq k \leq K\}$

3. Predict $\hat{y}_0^{(kNN)} = \arg\max_{y \in \mathcal{Y}} |\mathbf{x}'_y|$, breaking ties randomly.

2.2.2 Random Forest

Random Forest (RF) is an ensemble and supervised learning technique. RF was developed by [Breiman \(2001\)](#) that utilize decision tree as base classifier and generate multiple decision trees. Decision tree learning is an approach commonly utilized in data mining ([Rokach & Maimon 2008](#)). The goal of the decision tree is to construct a model, based on several training input variables that predicts the value of a target variable. In the RF algorithm, the individual trees need to be randomized to de-correlate the predictions. The most common method of randomization is the bootstrap aggregation also known as *bagging* ([Breiman 1994](#)) where a random selection of the training dataset is trained for each decision tree.

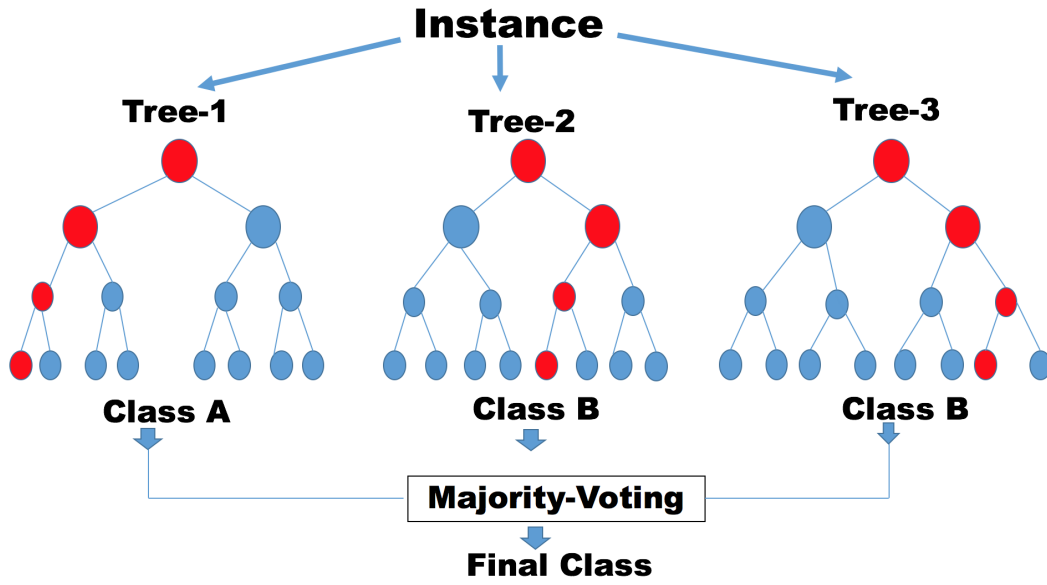


Figure 2.3: The majority voting across all trees is taken as the final class for classification using Random Forest.

RF is a classifier which consists of a collection of tree-structured classifiers where each tree casts a unit vote for the most popular class for input \mathbf{x} ([Breiman 2001](#)). Let $\{\mathcal{T}_m, \theta_m\}_{m=1}^M$ represent an ensemble of trees, where M is the number of trees in the ensemble. Let $g(\mathbf{x}; \mathcal{T}_m, \theta_m)$ denote the prediction from the m^{th} decision tree for a test data point \mathbf{x} .

The final prediction is simply the average of the predictions of individual trees which is given in Equation 2.4.

$$g(\mathbf{x}; \{\mathcal{T}_m, \theta_m\}_{m=1}^M) = \sum_{m=1}^M \frac{1}{M} g(\mathbf{x}; \mathcal{T}_m, \theta_m) \quad (2.4)$$

However, for classification, if the individual tree give as output discrete class labels instead of probability distributions, then it is possible to take the majority voting as shown in Figure

2.3.

The main disadvantage of RF is that the algorithm can be computationally expensive depending on the number of decision trees. However, RF performs efficiently on large datasets (Breiman 2001). Thousands of input variables can be handled without variable deletion when RF is implemented. In addition, RF ranks the features by their importance. RF also calculates an internal unbiased estimate of generalization error as forest growing progresses (Breiman 2001). On a concluding note, RF is a popular algorithm used in many classification and prediction problems (Kulkarni & Sinha 2013).

2.2.3 Naive Bayes Classifier

Bayesian reasoning is focused mostly on the theory that the quantities of interest are ruled by probability distributions. Optimal decisions can be made while working with these probabilities together with observed data. It is of importance to machine learning by focusing on the fact that it can be utilized for supplying a quantitative approach to weighing the evidence for various hypotheses. Bayesian reasoning therefore gives the pipeline for algorithms to directly learn and manipulate probabilities.

Naive Bayes (NB) focus on class conditional estimation and the class prior probability where the posterior class probability of a data point in the test set can be derived and the test data will be given to the class with the maximum posterior class probability (Ren et al. 2009). An introduction to Bayes theorem is presented in Appendix A.

In some fields, the performance of NB has been demonstrated to be comparable to neural networks. As seen in Appendix A, the Bayes theorem involves conditional and marginal probability of random events. It is used to calculate the posterior probabilities given observations. Let $a = (a^1, a^2, \dots, a^d)$ having no class label, be a d -dimensional instance. Our aim is to construct a classifier based on Bayes theorem to predict the unknown class label. The set of the class label is denoted by $C = \{C_1, C_2, \dots, C_K\}$. The prior probability $P(C_k)$ of C_k ($k = 1, 2, \dots, K$) are deduced before new evidence. The conditional probability of observing a if the hypothesis is correct, is written as $P(a|C_k)$. Bayes theorem is then used to construct a classifier

$$P(C_k|a) = \frac{P(a|C_k)P(C_k)}{\sum_k P(a|C_k)P(C_k)} \quad (2.5)$$

The naive Bayes classifier considers that the value of a specific feature of a class is unrelated to the value of any other feature such that

$$P(a|C_k) = \prod_{i=1}^d P(a^i|C_k) \quad (2.6)$$

where the superscript i is used on the multi-dimensional quantities to show their values in the i -th dimension. The main advantage of NB classifier is that it is fast to train and performs classification rapidly as well as being insensitive to irrelevant features. However, it assumes independence of features which is a drawback.

2.2.4 Multi Layer Perceptron

The Multi-Layer Perceptron (MLP) is an effective and powerful algorithm for solving supervised learning problems. Artificial neurons are a simplified version of biological neurons showing many interesting and useful properties in the research field. The starting point in Artificial Neural Network (ANN) was performed in 1943 by Warren McCulloch and Walter Pitts where they studied networks of *binary threshold elements*.

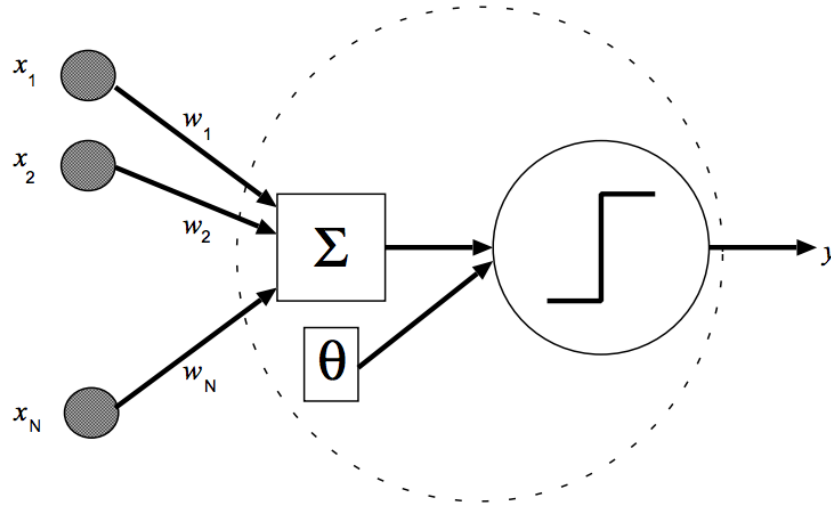


Figure 2.4: McCulloch-Pitts neuron where x_i is the input to the neuron and w_i is the weight associated to each input. θ is the threshold to the Heaviside step function and Σ is the weighted sum of the inputs. Figure courtesy of [Gallagher \(1999\)](#).

Figure 2.4 illustrates the McCulloch-Pitts neuron where each artificial neuron or unit has N inputs x_i which is associated with a *weight* value w_i ([Gallagher 1999](#)). The artificial neuron calculates a function of the weighted sum of its inputs and gives a value of 0 if the sum is below some threshold, θ and a value of 1 otherwise, where g is called the Heaviside step function as given in Equation 2.7.

$$g(b) = \begin{cases} 1 & \text{if } b - \theta > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

where b is the weighted sum of the inputs given as

$$b = \sum_{i=1}^N x_i w_i \quad (2.8)$$

Various thresholds also known as *activation functions* are used in ANN. If one wants a symmetric output from the ANN, the Heaviside function g is replaced by a step function (also known as signum)

$$\text{sgn}(b) = \begin{cases} 1 & \text{if } b - \theta > 0 \\ -1 & \text{otherwise} \end{cases} \quad (2.9)$$

Another types of activation function are the *sigmoidal function*, for example the logistic function

$$f(b) = \frac{1}{1 + e^{-b}} \quad (2.10)$$

and the hyperbolic tangent function

$$\tanh(b) = f(b) = \frac{e^b - e^{-b}}{e^b + e^{-b}} \quad (2.11)$$

2.2.4.1 The Architecture of the Multi Layer Perceptron

Figure 2.5 illustrates the MLP architecture where signal flows from the input layer with N_i neurons (and an input bias neuron) passing through a single hidden layer with N_h neurons and finally to the output layer of nodes, N_o . There can be one or more hidden layers. Inside MLP, different activation functions $f_o(\cdot)$, $f_h(\cdot)$ can be used for the hidden and output layers depending on the problem. Each neuron is fully connected to the neurons of the layers above and below it. Therefore, information flows from the input layer towards the output layers in one direction and this network is known as *Feed-Forward Neural Network*. The network mapping is $y = \phi(x, w)$ where the input is denoted as x , the output as y and w as the weights.

MLP performs a mapping from the input space to the output space by performing a non-linear transformation given by the weights and the activation functions of the network. The

weights of the network are parameters that are adjusted during the training phase. MLP optimization requires tuning of the hyperparameters, for instance the number of hidden neurons, hidden layers and number of iterations, all of which impact the validation accuracy.

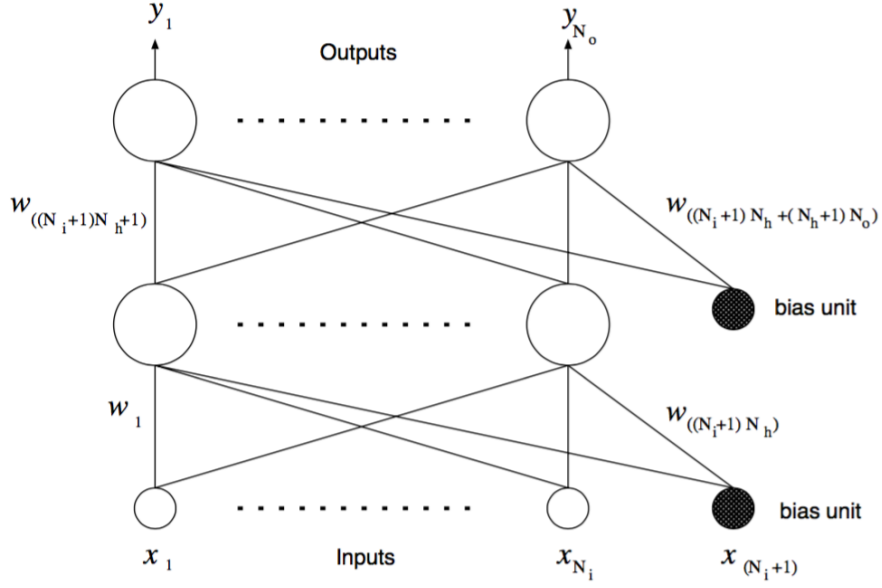


Figure 2.5: The architecture of MLP. Signal flows from the input layer with N_i neurons through the hidden layers with N_h neurons and finally to the output layer with neurons N_o . Figure from [Gallagher \(1999\)](#).

2.3 Dimensionality Reduction

Dimensionality reduction is useful when we have data with many dimensions, for example a multi-pixel image of a face or texts from an article. Describing them in simpler way will allow us to analyse them in a simpler fashion. Therefore, the unsupervised learning technique of dimensionality reduction, in particular the Principal Component Analysis (PCA) is discussed.

PCA also known as the Karhuen-Loeve transform, is an approach generally employed for applications, in particular, data compression and visualization, dimensionality reduction and feature extraction ([Bishop 2000](#)). [Hotelling \(1993\)](#) defined PCA as a set of data being projected orthogonally onto lower dimensional linear space so that the projected data's variance is maximized. This new linear space with lower dimension is called the principal subspace. Equivalently, [Pearson \(1901\)](#) designated PCA as the linear projection that reduces the mean squared distance between the original data points and their projections are known as the average projection cost ([Bishop 2000](#)).

The main goal of PCA is to reduce the dimensions of a data set of many correlated variables such that most of the variance existing in the original data set is preserved in the modified

data set. This process is done by orthogonally transforming the original data to a new set of uncorrelated variables, also known as *principal components* (PCs). The first few PCs are arranged in such a way that they contain most of the variance of the original data set (Jolliffe 2002). The algebraic derivation of PCA is specified in Appendix B.

2.4 Application of Machine Learning in Astronomy

Nowadays, machine learning techniques are applied to different problems in astronomy, particularly for automatic identification of galaxy morphologies (Dieleman et al. 2015), star-galaxy classification (Kim & Brunner 2016), estimation of redshift (Hoyle 2016), and identification of transients (Goldstein 2015). The learning algorithms that are mostly used are Naive Bayes (NB), k Nearest Neighbours (k NN) and decision trees. These algorithms learned on features extracted from data. It is important to perform feature extraction carefully as these features will correspond to specific physical information and properties of the original data. The accuracy of the algorithm generally depends on the quality of features extracted from the system. This type of network is referred to as shallow learning (Chen 1995) which learned from features rather than from raw data.

Star-Galaxy classification is one of the most studied applications of machine learning in optical astronomy. In this work, the aim is to perform radio galaxy classification. Previously, source detection and classification was performed in a traditional way by visual inspection and through various source finding software packages as discussed in Section 1.4. However, this is infeasible with the increase in survey sizes hence, the need to adopt automated techniques. Banfield et al. (2015) used crowd sourcing on Radio Galaxy Zoo images, Proctor (2016) applied pattern recognition and decision trees, Van Velzen et al. (2015) utilized source matching and pattern recognition and Polsterer et al. (2015) used self-organizing maps which is popular in the application of pulsar and transient detection.

In addition, star-galaxy separation has been tackled using neural networks (Odewahn et al. 1992, Odewahn & Nielsen 1994, Connolly et al. 1995, Bazell & Peng 1998, Andreon et al. 2000, Philip et al. 2002, Odewahn et al. 2004) and decision trees (Weir et al. 1995, Ball et al. 2006) that are implemented using algorithm inputs of morphological parameters derived from photometry survey. These algorithms have achieved a precision score (see Section 4.5.2) of around 95%. Distinguishing galaxy morphologies has also been investigated. Information about evolution and formation of galaxies can be obtained from their shapes and sizes. Storrie-Lombardi et al. (1992), Connolly & Szalay (1999) applied neural networks using measured parameters such as

morphological parameters and color information and they found that the accuracy is nearly as high as that of human experts. Madgwick (2003) used galaxy spectra as input to neural networks to predict morphological types of galaxies. Previously, it was difficult to perform classification of galaxy morphology since galaxies at higher redshift are fainter, distant, less evolved and more peculiar. Nowadays, using the Hubble Deep Field data, galaxies are classified using neural networks with surface brightness and light profiles of the galaxies as input to the algorithms (Odewahn et al. 1996, Windhorst et al. 1999, Cohen et al. 2003).

Spectral classification of galaxies has been achieved using principal component analysis (Connolly et al. 1995, Connolly & Szalay 1999, Madgwick et al. 2001, Yip et al. 2004). Another important and well studied object, is the identification of quasars and AGN as they are fundamental tools to a deeper knowledge about the evolution and formation of structure in the universe. Many studies combine multi-wavelength data to select quasars from surveys where Carballo et al. (2004), Claeskens et al. (2006), Carballo et al. (2008) utilized neural networks and White (2000), Zhang & Zhao (2007), Knigge et al. (2008) implemented decision trees.

Waisberg (2013) applied machine learning algorithm for the classification of astronomical point sources into main sequence/red giant stars, white dwarfs or quasars. Photometric data from the SDSS survey across 5 bands are used as features for the supervised classification. It was found that k NN and multinomial SVM performed best.

Riggi et al. (2016) presented a new algorithm known as CAESAR (Compact And Extended Source Automated Recognition) to detect extended sources in radio maps. They apply a pre-filtering technique to denoise the images. Afterwards, a compact source suppression and enhancement of diffuse emission are implemented which is then followed by adaptive superpixel clustering for the final segmentation of the sources. It is found that the designed algorithm is able to detect known target sources and regions of diffuse emission.

Kim & Brunner (2016) presented a star–galaxy classification pipeline that utilized a convolutional neural network (ConvNet) model directly on the images from the SDSS and the Canada–France–Hawaii Telescope Lensing Survey (CFHTLenS). They compared the performance of the model with a standard machine learning technique that uses the reduced summary information from catalogues. They demonstrated that their ConvNet model produced accurate and well-calibrated probabilistic classifications.

Aniyan & Thorat (2017) applied convolutional neural networks to classify radio images on a morphological basis. In their study, they classified FRI, FRII and bent-tailed radio galaxies with respective individual precision of 95%, 91% and 75%. They utilized a fusion model to perform

classification of the test data where they successfully classified FRI and FRII radio galaxies with a F1 score (see Section 4.5.2) of 86%.

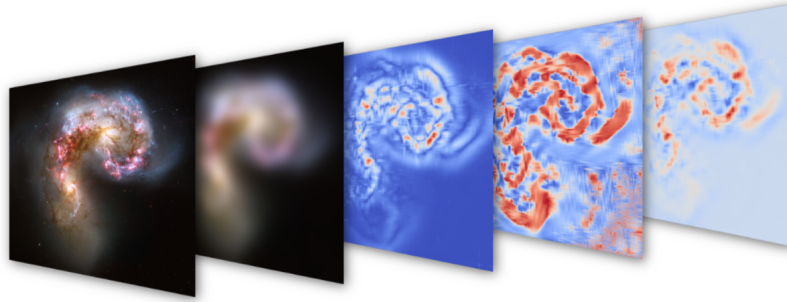


Figure 2.6: Starting from left to right is the original image of a galaxy merger, the scale-space representation of the galaxies, the curvedness (a measure of how pronounced the local structure is), the shape index, and finally the shape index weighted by the curvedness. Figure from Kremer et al. (2017).

Kremer et al. (2017) illustrated three examples where they used machine learning and image analysis on astronomical data. Firstly they used shape index to predict galaxy evolution and star formation rate as shown in Figure 2.6. The shape index calculates the local structure around a pixel starting from dark blobs passing over valley-saddle point and ridge-like structures to white blobs (Kremer et al. 2017).

Nowadays, it is observed that machine learning and deep learning are emerging in astronomy with applications in time domain data, in imaging for classification of galaxies, asteroids and other astronomical source classification. Also, various parameters of the stellar atmosphere can be used to make predictions that will help in shedding light on the evolution of the universe.

2.5 Summary

The machine learning algorithms and dimensionality reduction techniques presented in this chapter provide the theoretical background for later chapters. Background on the random forest classifier, naive Bayes classifier, the k-nearest neighbours algorithm, the multi-layer perceptron will be necessary for the discussions in Chapter 4. In addition, the application of machine learning and deep learning in astronomy was discussed.

Chapter 3

3 Astronomical Source Detection using Filter-based Methods

In this chapter, the aim is to identify extended sources in astronomical images. Following the review of traditional methods used in the past, some of those methods mentioned in Section 1.4 are implemented in this chapter. For example the local peak search (see Section 1.4.2.2), thresholding (see Section 1.4.2.1) are implemented to detect sources and to extract sources in radio images. In addition, we provide the first application of LULU operators to radio astronomy. It is important to firstly identify the sources in the data before gaining photometry and morphological measurements of astronomical sources. The Photutils⁴ package is used, which is an astropy package developed by Bradley et al. (2016). A sample of around 151 astronomical images is used from the SUMSS survey, having both point-like and extended sources. In this chapter, the image 2MASX J02581124-5243419 is used to illustrate the results obtained. The theoretical number of sources in this image is found using Vizier⁵, a library of published astronomical catalogues and data tables. It is found that this image consists of 7 sources with 3 point-like and 4 extended sources.

3.1 Local Peak detection

Peaks in an image are the local maxima that are separated by a specified minimum number of pixels above a specified threshold within a local region. The regions are determined by a box size parameter that defines the local region around each pixel as a square box. If within a local region, multiple pixels have the same intensities, then all the pixels coordinates that

⁴<https://photutils.readthedocs.io/en/stable/#>

⁵<http://vizier.u-strasbg.fr/>

are returned back are always integer-valued. On the other hand, only one pixel coordinate is returned for one peak pixel per local region.

Table 3.1: Peak Values of the sources in the 2MASX J02581124-5243419 image.

| X pixel location | Y pixel location | Peak value/mJy |
|------------------|------------------|----------------|
| 94 | 109 | 10.9 |
| 83 | 114 | 16.8 |
| 155 | 114 | 5.5 |
| 163 | 137 | 13.3 |
| 280 | 163 | 3.3 |
| 142 | 177 | 13.0 |
| 157 | 181 | 3.8 |
| 106 | 228 | 11.5 |
| 55 | 255 | 6.8 |
| 10 | 272 | 8.3 |
| 10 | 273 | 8.3 |

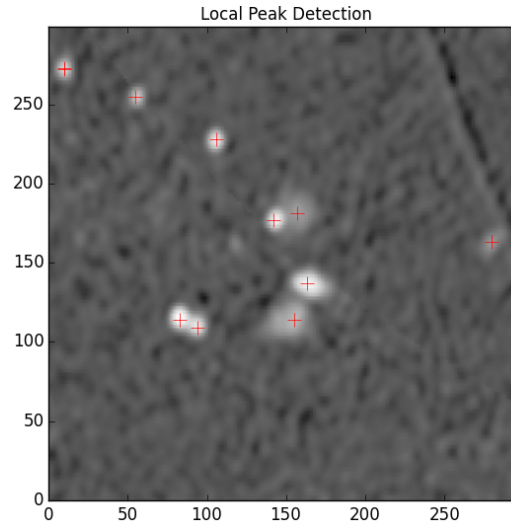


Figure 3.1: The location of the detected peaks (red cross) in the 2MASX J02581124-5243419 image. The algorithm has considered extended source with the following coordinates (150, 179), (159,126) and (89, 109) as two separate sources.

In Figure 3.1 only the peak pixels that are above 10 sigma, where sigma is the background RMS are returned. The peak values of the detected sources with their corresponding coordinates are given in Table 3.1. However, this is not a good method to find the actual number of sources in an image as a connected sources (extended sources) as seen in Figure 3.1 can have two peaks. The algorithm has eventually counted them as two separate sources. For exam-

ple, ancillary data from Vizier has classified sources with the following coordinates (150, 179), (159,126) and (89, 109) as extended sources (that is, two radio components associated with a single galaxy). However, these sources at these coordinates (150, 179), (159,126) and (89, 109) obtained from Vizier, are considered as two separate sources by the local peak algorithm (that is, the two radio components are associated with two different galaxies).

3.2 Source Properties

The concept of estimation in image processing is related to the evaluation of image parameters which is considered to be appropriate to describe the characteristics of the sources in the image. Therefore, the morphological properties of the detected sources can be measured in the image analysis which include source sum, area, centroid and many more. For each sources in the image, the photometry and morphological properties of sources are calculated as illustrated in Table 3.2.

Table 3.2: Properties of the sources present in the 2MASX J02581124-5243419 image.

| ID | xcentroid pixel location | ycentroid pixel location | Source Sum /mJy | Source Sum Error /mJy | Area/pixel ² | Background at centroid /mJy | Background Mean /mJy |
|----|--------------------------------|--------------------------------|-----------------------|-----------------------------|-------------------------|-----------------------------------|----------------------------|
| 1 | 87.1 | 113.0 | 2.48 | 0.071 | 371.0 | 0.034 | 0.034 |
| 2 | 159.6 | 127.2 | 4.03 | 0.090 | 850.0 | 0.024 | 0.025 |
| 3 | 279.5 | 162.7 | 0.23 | 0.022 | 95.0 | 0.005 | 0.005 |
| 4 | 149.0 | 178.7 | 1.86 | 0.061 | 456.0 | 0.053 | 0.051 |
| 5 | 105.4 | 226.7 | 0.83 | 0.041 | 156.0 | 0.069 | 0.069 |
| 6 | 55.8 | 254.1 | 0.41 | 0.029 | 104.0 | 0.066 | 0.066 |
| 7 | 10.0 | 271.9 | 0.56 | 0.034 | 128.0 | 0.062 | 0.062 |

- The source sum is calculated using Equation 3.1.

$$F = \sum_{i \in S} (I_i - B_i) \quad (3.1)$$

where F is the source sum, I_i is the i^{th} pixel value of the image, B_i the i^{th} pixel value of the background and S are the non-masked pixels in the source segment.

- The source sum error is the quadrature sum of the total errors over the non-masked pixels within the source segment as given in Equation 3.2.

$$\Delta F = \sqrt{\sum_{i \in S} \sigma_{tot,i}^2} \quad (3.2)$$

where ΔF is the source sum error, $\sigma_{tot,i}$ are the pixel-wise total errors and S are the non-masked pixels in the source segment.

- The area is simply the area of the source segment in units of pixels².
- The background at the centroid is the value of the background at the position of the source centroid. Fractional position values are determined using bilinear interpolation.
- The background mean is the mean of background values within the source segment.

3.3 Centroids

In optical images, stars can be considered as point sources. However, a source whose size is above a certain area is no longer a point. Therefore, a precise meaning is given to the word ‘position’ of a star. So the coordinate/position of an astronomical source can be determined from its centre of area which is estimated by its centroid or centre of mass. In addition, the position of a source can also be estimated by using its maximum intensity but this gives the accuracy to only one pixel. So, it is preferable to use the centre of mass instead (Eisfeller & Hein 1994, Buil 1991). There are various techniques to find the centre of mass and these are illustrated as follows: (i) A centroid detection algorithm is constructed by using Gaussian pattern matching (Vyas et al. 2010). (ii) Also, in the Deep Sky Object (DSO), a method for detecting source’s centroid is proposed which is based on the generation of an error minimization signal (Suszynski & Wawryn 2015). However, the basic technique used to estimate the centre of mass of a source is the image moment analysis.

3.3.1 Image Moments

When a sample of values is clustered around some specific value, then it can be used in a useful way to characterize the sample using only a few numbers that have a relationship with its moments, that is, the sums of integer powers of the values (Suszynski & Wawryn 2015). In an image, a source can be defined as $I(x, y)$, then features can be generated from the moments. From Papoulus thorem (Gonzalez & Wintz 1987), the $(K + L)^{th}$ order is defined for digital images by Equation 3.3.

$$I_{KL} = \sum_x \sum_y x^K y^L I(x, y) \quad (3.3)$$

The total intensity of the image is denoted by I_{00} and we found that moments depend on the

intensity level. Higher-order moments have a dependency on the total intensity. With image moments, we can find the centre of mass, the variance and orientation of an object.

3.3.2 Centre of Mass

If we consider the intensity $I(x, y)$ at each point (x, y) of an image I as the mass of (x, y) , then, the centroid and other moments can be defined at the place where all the mass of an object is accumulated. So, the centre of mass of the source can be defined as the centre of area of the source which is chosen at the position of the source (Suszynski & Wawryn 2015). If we consider a two dimensional situation, (I_{10}, I_{01}) is the centre of mass. To calculate the centroid of a binary image you need to calculate two coordinates as given in Equation 3.4.

$$Centroid = \left(\frac{I_{10}}{I_{00}}, \frac{I_{01}}{I_{00}} \right) \quad (3.4)$$

Consider the first moment

$$I_{10} = \sum \sum xI(x, y) \quad (3.5)$$

The two summations are like a *for* loop. The x coordinate of all white pixels (where $I(x, y) = 1$) is added up. Similarly, we calculate the sum of y coordinates of all white pixels.

$$I_{01} = \sum \sum yI(x, y) \quad (3.6)$$

Now we have the sum of several intensities in x and y coordinates. To get the average, we divide each by the total intensity I_{00} .

$$I_{10} = \frac{\sum \sum xI(x, y)}{I_{00}} \quad (3.7)$$

$$I_{01} = \frac{\sum \sum yI(x, y)}{I_{00}} \quad (3.8)$$

Using the 2MASX J02581124-5243419 image, firstly each source is extracted from the image and the centroid for each individual source is found using three different methods. Figure 3.2 shows the original 2MASX J02581124-5243419 image in the middle and each source is extracted showing the centroid in zoomed plots. The ‘Black plus’ represents the centre of mass of the sources calculated from the 2-D Moments. The ‘purple plus’ represents the centroid obtained by fitting 1-D Gaussian to the marginal x and y distributions of the data. The ‘red plus’ is the

centroid calculated by fitting a 2D-Gaussian to the 2D distribution of the data. From literature review, these methods are used to find the centroid of sources.

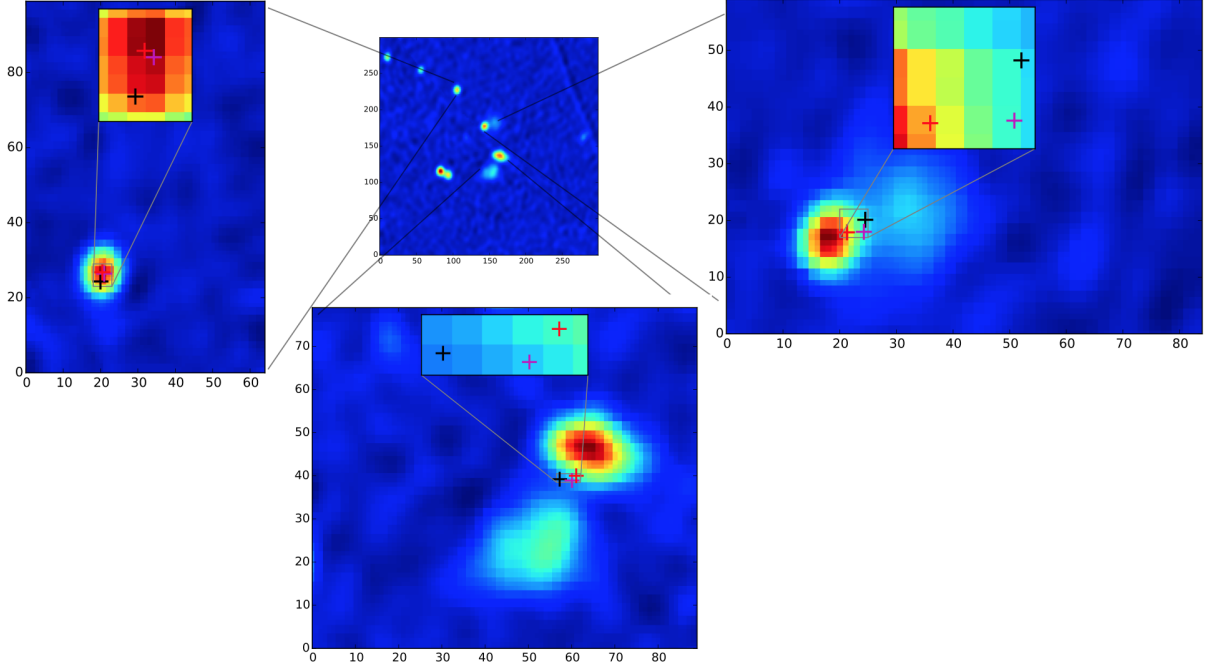


Figure 3.2: Centroids of different sources in the 2MASX J02581124-5243419 image of the SUMSS data using three methods. The centre of mass of the sources is represented by 'plus' marks where the 'black plus' is calculated from 2-D moments, 'purple plus' is obtained by fitting 1-D Gaussian to the x-y distribution of the source and the 'red plus' is done by fitting a 2D-Gaussian to the 2D distribution of the data.

3.4 Source Extraction Using Image Segmentation

In computer vision, sources (both point-like and extended) can be detected in an image using the process of image segmentation called the thresholding method (see Section 1.4.2.1). In this case, detected sources consist of a minimum number of connected pixels that are each greater than a certain fixed threshold value which is usually stipulated at some multiple of the background standard deviation. The method used to extract the sources from the image, is explained as follows:

3.4.1 Methodology

1. A label is assigned to every pixel in the image in such a way that pixels with the same label form part of the same source.
2. Before thresholding, to smooth the noise and maximize the detectability of objects, the image can also be filtered using a 2D circular Gaussian kernel. The downside of the filtering is that sources will be made more circular than they actually are.

3. For the background and noise estimation, the most widely used technique to remove the sources from the image statistics is called sigma clipping where the pixels that are above or below a specified sigma level from the median are discarded and the statistics are recalculated. The procedure is typically repeated over a number of iterations or until convergence is reached. This method provides a better estimate of the background and background noise levels
4. After applying the simple sigma-clipped statistics to estimate the background and background rms, a 2D detection threshold image is generated, where detected sources have a minimum number of connected pixels that are each greater than a specified threshold value in an image.
5. To avoid overlapping sources to be considered as a single source, a deblending procedure is implemented. It is based on a combination of multi-thresholding and watershed segmentation. To successfully deblend the sources, they must be separated enough in such a way that there is a saddle between them.

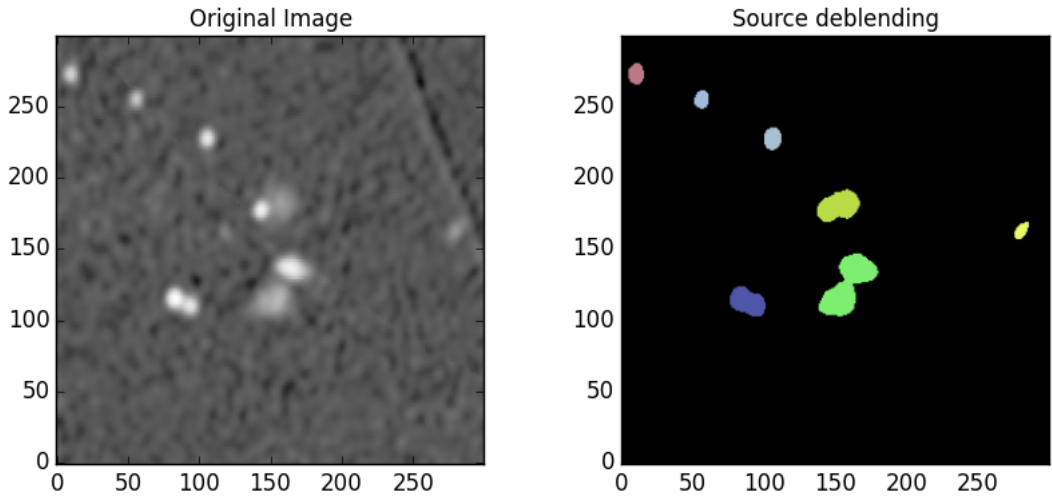


Figure 3.3: The left panel illustrates the 2MASX J02581124-5243419 image and the right panel shows the deblended segmentation image where the different colours are positive integers assigned to each source.

Figure 3.3 shows the final deblended segmentation image where a value zero is reserved for the background and the different colours represent the sources labelled by different positive integer values.

Our main aim is to detect only the 4 extended sources present in the original image, however, we can observe that both point-like and extended sources are successfully extracted from the image. We will therefore employ different strategies in extracting extended sources, for

instance using a Discrete Pulse transform along with LULU operators (Rohwer 2005) and we also implement some filtering techniques to compare different cases.

3.5 Discrete pulse transform of images and applications

During the last three decades, the LULU operators and the Discrete Pulse Transform (DPT), were developed and implemented in the field of signal processing analysis. Its recent extension into higher dimensions paved the way for applications on signals such as images. The DPT extracts discrete pulses of the image of every possible shape and size thereby setting the stage for an effective computer vision method. The practical soundness of the DPT is investigated in image sharpening, best approximation of an image, noise removal in signals and images, feature point detection with ideas to extending work to object tracking in videos, and image segmentation. This study also represents the first application of LULU and DPT to radio astronomy. Their implementation on radio images is aimed to reduce noise and to extract meaningful sources.

All real astronomical images are affected by noise, therefore noise removal is a fundamental aspect of image processing. In addition, the goal of the work is to extract extended sources from radio images, therefore an image segmentation process is also implemented. We can therefore define image segmentation as the division of an image into regions which match different objects or some parts of objects. In this work, we present the implementation of the two-dimensional LULU and Discrete Pulse Transform.

The one dimensional LULU theory and the DPT were developed originally by Rohwer (2005). Anguelov & Fabris-Rotelli (2010) developed the multi-dimensions of the LULU operators and the DPT. The LULU smoothers are given by the operators L_n and U_n which are the morphological filters that are operated on neighbourhoods of size n and the DPT is a new non-linear decomposition of a multidimensional array. The DPT of images is obtained via recursive peeling of so-called local maximum and minimum sets with the LULU operators as n increases from 1 to the maximum number of elements in the array.

Rahmat et al. (2013) have investigated the effectiveness of the Median filter as compared to the LULU filter for impulse noise removal in images. Impulse noise can be identified as black and white dots also known as salt and pepper noise on images. By noise in an image, it means that the color or brightness intensity spatially varies. This is due to various different effects, devices or image sensors that degrade the quality of images. Therefore, before analyzing any images the first step is the removal of noise. In the work of Rahmat et al. (2013), four different

LULU filters are compared with the median filter to address impulse noise removal which is done through two different image quality measures using the Root Mean Square Error (RMSE) and Peak Signal to Noise Ratio (PSNR) techniques. [Rahmat et al. \(2013\)](#) concluded that the median filter protected the edges as well as eliminated noise but however it is computationally less efficient than LULU methods.

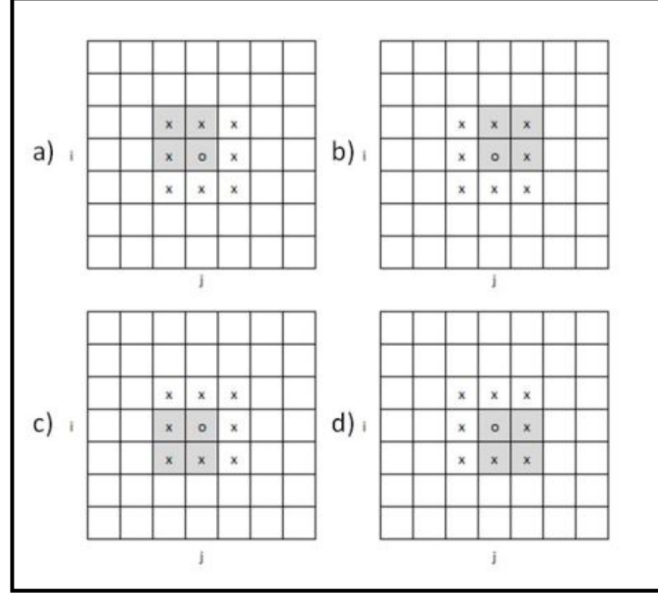


Figure 3.4: Illustration of neighbours for Equation 3.9 to Equation 3.12, a) I1, b) I2 c) I3, and d) I4. Figure courtesy of [Rahmat et al. \(2013\)](#).

In [Fabris-Rotelli \(2009\)](#), the computational theory of the LULU operators and the Discrete Pulse Transform (DPT) were presented in details. LULU is mostly the combination of two operators L (lower) and U (upper) and recently in [Anguelov \(2008\)](#), it is applied for the extraction of objects in images by applying a Discrete Pulse Transform (DPT). LULU operators can be mostly used for smoothing or filtering images. An explanation about applying LULU on images is given using the following example. If we consider the pixel $I(i, j)$ on an image and then the latter is segmented into four different parts as shown in Figure 3.4 and given in Equation 3.9 to Equation 3.12.

$$I_1 = [I(i-1, j-1), I(i-1, j), I(i, j), I(i, j-1)] \quad (3.9)$$

$$I_2 = [I(i-1, j+1), I(i-1, j), I(i, j), I(i, j+1)] \quad (3.10)$$

$$I_3 = [I(i, j-1), I(i, j), I(i+1, j-1), I(i+1, j)] \quad (3.11)$$

$$I_4 = [I(i, j + 1), I(i, j), I(i + 1, j), I(i + 1, j + 1)] \quad (3.12)$$

Then, the L and U operators are applied as shown in Equation 3.13 and Equation 3.14. To obtain LU filter, L is firstly applied on the image. Then on the result of L operator, U is applied.

$$L(i, j) = \max[\min(I_1), \min(I_2), \min(I_3), \min(I_4)] \quad (3.13)$$

$$U(i, j) = \min[\max(I_1), \max(I_2), \max(I_3), \max(I_4)] \quad (3.14)$$

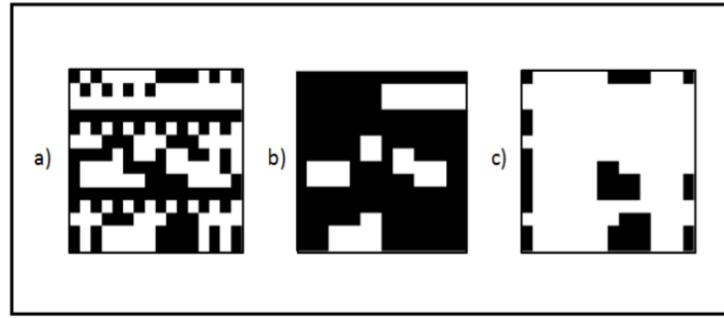


Figure 3.5: a) Image with impulse noise, b) Result of L smoother on the image, and c) Result of U smoother on the image. Figure courtesy of [Rahmat et al. \(2013\)](#)

Figure 3.5(a) illustrates a binary image with noise. L operator removes the lower peaks by extracting the minimums of the neighborhood, thus leaving more black marks compared to the original image as shown in Figure 3.5(b). While with the U smoother, more white marks are seen in Figure 3.5(c). This shows the basic mechanism about how the LULU operators are employed.

3.6 The Discrete Pulse Transform

The Discrete Pulse Transform is acquired by the successive removal of peaks (local maximum sets) and valleys (local minimum sets) from an image by applying L_n and U_n respectively. A more detailed explanation is given in [Fabris-Rotelli & Van der Walt \(2009\)](#).

Let B be an arbitrary non-empty set and a family C of subsets of B is called a connected class. Let $V \in C$ be a connected set. A point $V \notin C$, is called adjacent to V if $V \cup \{x\} \in C$. The set of all points adjacent to V is denoted by $\text{adj}(V)$, that is,

$$\text{adj}(V) = \{x \in \mathbb{Z}^2 : x \notin V, V \cup \{x\} \in C\}$$

A connected subset V of \mathbb{Z}^2 is called a local maximum set of $f \in \mathcal{A}(\mathbb{Z}^2)$ where $\mathcal{A}(\mathbb{Z}^2)$ is the vector lattice (admitting a supremum and infimum) of all real functions defined on \mathbb{Z}^2 if

$$\sup_{y \in \text{adj}(V)} f(y) < \inf_{x \in V} f(x) \quad (3.15)$$

Similarly, V is a local minimum set if

$$\inf_{y \in \text{adj}(V)} f(y) > \sup_{x \in V} f(x) \quad (3.16)$$

Figure 3.6 shows the local maximum set and local minimum set. The LULU operators operated as follows on local maximum and minimum sets.

- The application of $L_n(U_n)$ removes local maximum sets of size smaller or equal to n .
- When there is no local minimum or maximum sets, there will be no creation of new local minimum or maximum sets by the two operator L_n and U_n . However, the existing minimum or maximum sets might be enlarged when adjacent sets are joined.
- The condition $L_n(f) = f(U_n(f)) = f$ only exists if and only if f does not have local maximum (minimum) sets of size n or less.
- Both the $(L_n \circ U_n)(f)$ and $(U_n \circ L_n)(f)$ have neither local maximum sets nor local minimum sets of size n or less where \circ is the Hadamard product. In addition, the condition $(L_n \circ U_n)(f) = (U_n \circ L_n)(f) = f$ only exists if and only if f does not have local maximum sets or local minimum sets of size less than or equal to n .

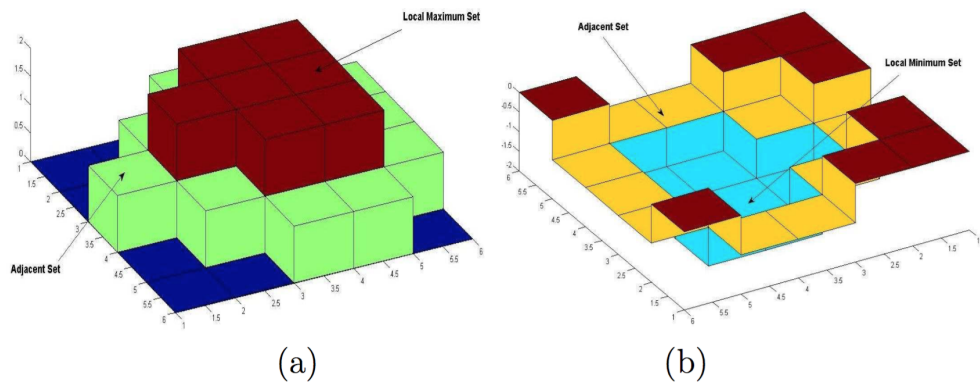


Figure 3.6: (a) Local Maximum set; (b) Local Minimum set. Figure adapted from [Fabris-Rotelli \(2009\)](#).

Now, consider $N = \text{card}(\text{supp}(f))$, that is, the size of the image where card is the cardinality of a set and supp is the support of a function. The DPT of $f \in \mathcal{A}(\mathbb{Z}^2)$ is derived by applying

iteratively the operators L_n, U_n with n increasing from 1 to N as follows:

$$DPT(f) = (D_1(f), D_2(f), \dots, D_N(f)) \quad (3.17)$$

where the components of Equation 3.17 are obtained through

$$D_1(f) = (I - P_1)(f)$$

where I is the identity operator,

$$D_n(f) = (I - P_n) \circ Q_{n-1}(f), n = 2, \dots, N$$

and $P_n = L_n \circ U_n$ or $P_n = U_n \circ L_n$ and $Q_n = P_n \circ \dots \circ P_1, n \in \mathbb{N}$

3.6.1 Extraction of Extended sources using the LULU operator and the DPT.

The LULU operators together with the DPT are implemented to extract extended sources from the SUMSS dataset. A pseudo-code is used in this work for implementing LULU and DPT which is available from the website: Fast implementation of the Discrete Pulse Transform (LULU-operators) in 2D⁶. The code is used on the whole dataset and here only the image 2MASX J02581124-5243419 will be illustrated as an example.

In image processing, thresholding is usually the first step to pre-process images to extract objects of interest. Images are partitioned into two categories, that is, the gray levels of pixels belonging to the source are entirely different from the gray levels of the pixels belonging to the background of the image (Senthilkumaran & Vaithegi 2016). Therefore, thresholding can be considered as a simple but effective method to partition those foreground objects from the background. Previously, various thresholding techniques have been proposed using global and local techniques. Only one threshold is applied to the entire image for global methods while different threshold values are applied to different regions of the image for local thresholding methods. The value is determined by the neighborhood of the pixel to which the thresholding is being applied (Leedham et al. 2003).

Consider a single threshold, t and in many situations, t is chosen manually by attempting a range of values of t and observing which one of the thresholding performs best at identifying the objects of interest. Therefore, in this work, a thresholding method is applied where different t based on the image is used followed by the application of the LULU operators

⁶<https://github.com/stefanv/lulu>

and the DPT. So, here the LULU is aimed at removing impulse noise. Then, DPT is applied which resulted into a number of resolution layers, that is, it decomposed the image into a collection of pulses. The image is now separated into its noise components and its true components where meaningful features are found. Afterwards, a low and high pass filtering is applied to the decomposed image selecting only extended sources with a certain area, that is, $\text{minimum area} \leq \text{Extended Sources} \leq \text{maximum area}$. Finally, an image mask is created with all sources having pixel value 1 and the background is assigned a value of zero. Since we aimed at extracting only extended source in the images, different thresholds are applied as illustrated in the various cases.

- **Case 1: A thresholding of $t = \text{mean}(I) + 3\sigma$**

After applying the LULU decomposition and the DPT with a thresholding $t = \text{mean}(I) + 3\sigma$ where I is the original image, Figure 3.7 is generated. It is observed that one extended source at position (175,125) is extracted as two separated sources. Therefore, this thresholding does not work well.

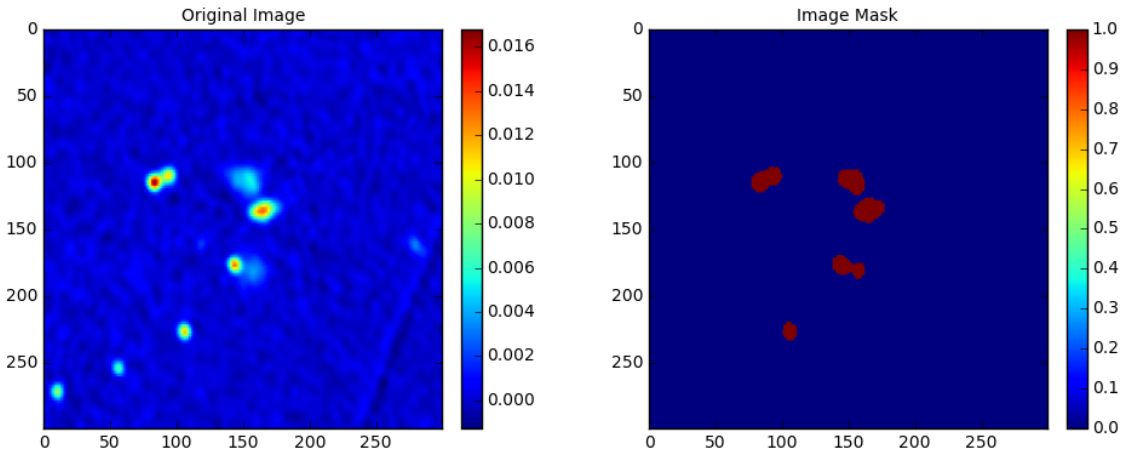


Figure 3.7: 2MASX J02581124-5243419 image thresholded at $t = \text{mean}(\text{Original image}) + 3\sigma$.

- **Case 2: A thresholding of $t = \text{median}(I)$**

The median value of the image is taken as the thresholding in this case. In Figure 3.8 both point-like and extended sources are picked up.

- **Case 3: A thresholding of $t = \text{median}(I) + 3\sigma$**

With the application of this thresholding in case 3, from Figure 3.9, we observed that one extended source is extracted as two separate sources.

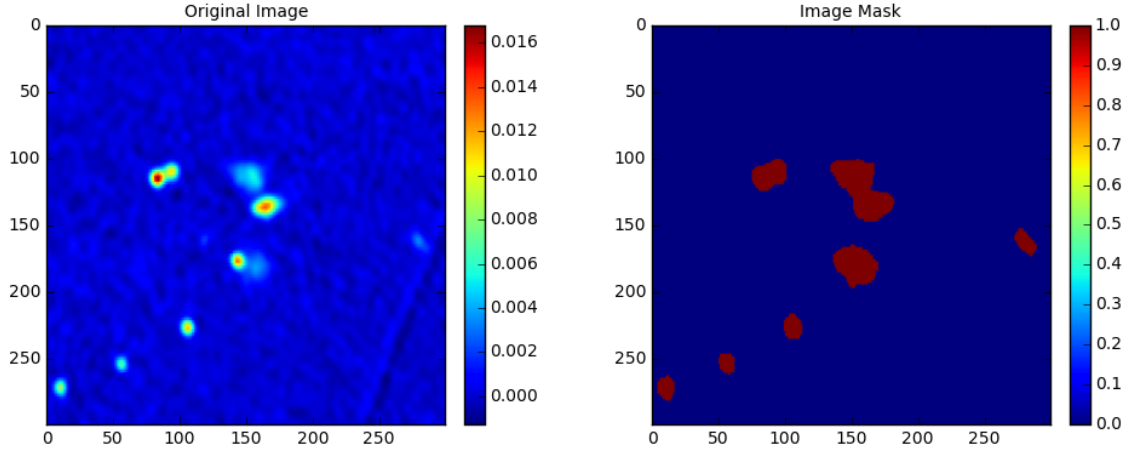


Figure 3.8: 2MASX J02581124-5243419 thresholded at the $t = \text{median}(\text{image})$.

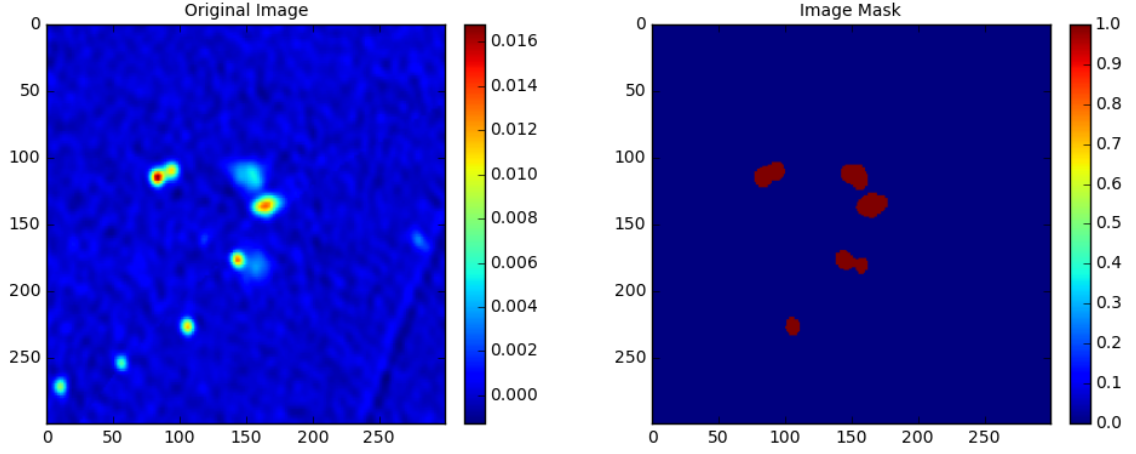


Figure 3.9: 2MASX J02581124-5243419 thresholded at the $t = \text{median}(\text{image}) + 3\sigma$.

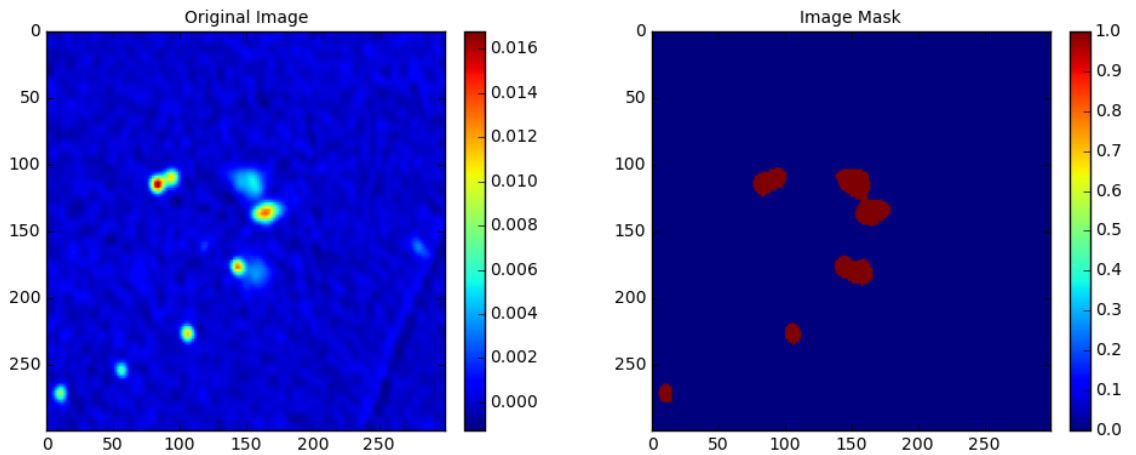


Figure 3.10: 2MASX J02581124-5243419 thresholded at the $t = 1\sigma$.

- **Case 4: A thresholding of $t = 1\sigma$**

The threshold is set to 1-sigma noise limit, that is, 1 mJy. Both extended and point sources are

segmented as illustrated in Figure 3.10.

- **Case 5: A thresholding of $t = \text{rms}(I)$**

The threshold is set to the noise of the image, that is, it is set to the rms of the original image shown in Figure 3.11.

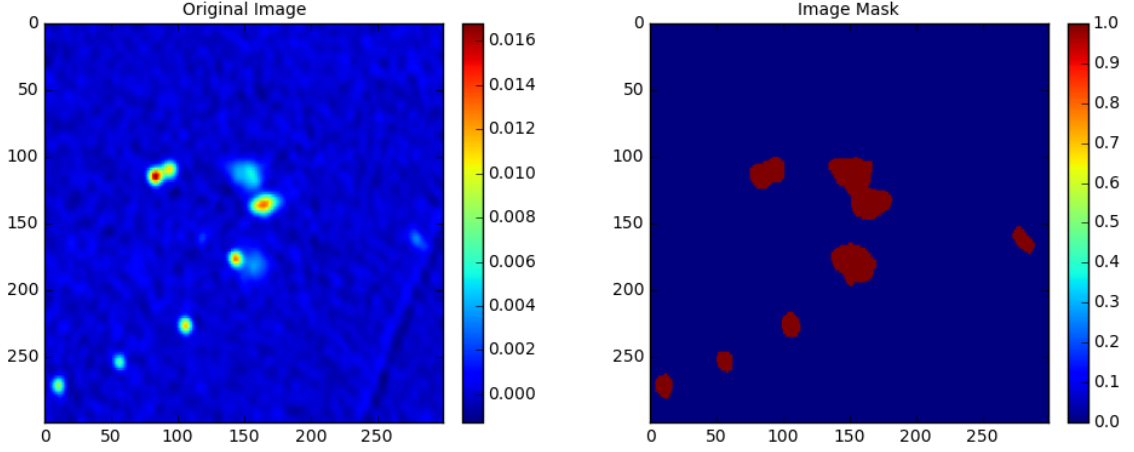


Figure 3.11: 2MASX J02581124-5243419 thresholded at the noise level of the image.

Five cases of thresholding were implemented on the image and we found that *Case 1* and *Case 3* have been successful in extracting extended sources as seen in Figure 3.7 and Figure 3.9. However, for both cases (1 and 3), one extended source is extracted as two different sources. Also, this method of LULU and the DPT are seen to pick noise and artefacts or point sources together with the extended sources. Since the DPT is a fairly new mathematical framework, it is prone to leakage within the domain. Leakage can be defined as unwanted union of two connected sets that gives false connectedness information regarding the data and it occurs due to over-segmentation (Stoltz & Fabris-Rotelli 2015). When *Case 1* is applied on the SUMSS dataset, the code took around 8 hours to extract extended sources in only 151 images. Therefore, this is not computationally feasible and has not been successful in extracting connected extended sources perfectly. Hence, another approach that incorporates some filtering approach is implemented in Section 3.7.

3.7 Extracting extended sources by image segmentation method using filtering techniques

A simple idea is to utilize the gray values of pixels to be able to separate sources from the background. Sources and blobs have higher intensities compared to the background. First, we

will perform some thresholding. For an automatic computation of the thresholding values, we will employ the Otsu's thresholding. It is an algorithm that chooses the threshold in such a way to have a good separation between gray values of background and foreground. The concept of Otsu thresholding is briefly elaborated in Appendix C.

3.7.1 Results of Otsu thresholding

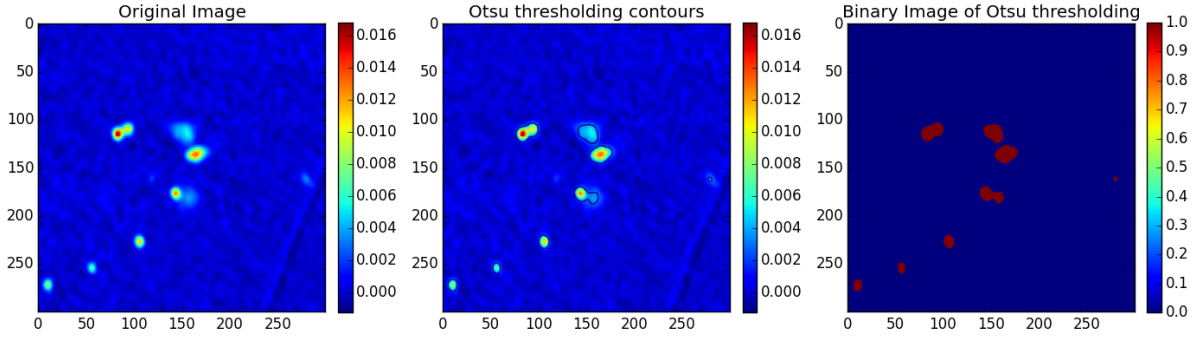


Figure 3.12: Application of Otsu thresholding on the 2MASX J02581124-5243419 image. The original image is plotted in the left panel, the result of otsu thresholding is shown in the middle and the binary image after the application of otsu thresholding is illustrated in the right panel.

Otsu thresholding is applied as a method to segment the image. A python package known as scikit-image developed by Van der Walt et al. (2014), which is dedicated to image processing and scikit-learn which is a simple and efficient tool for machine learning and data analysis developed by Pedregosa et al. (2011), are utilized to apply the Otsu Thresholding.

Figure 3.12 illustrates the application of Otsu thresholding on 2MASX J02581124-5243419 image. The left panel is the original image, the middle plot shows the contours thresholding on the original image and the right panel illustrates the binary image after thresholding. It is noted that here as well, point sources are extracted together with the extended sources. Therefore, we need to improve this thresholding by applying some image filtering.

3.7.2 Image Filtering

Image filtering is fundamentally important to reduce noise, sharpen contrast, highlight contours and detect edges. Image filtering can be classified as linear and non-linear filtering. Linear filtering involves the convolution filters since they can be represented by matrix multiplication while non-linear filters are thresholding, image equalization and median filter operations. For this work, we will implement the Gaussian filtering and then implement Otsu thresholding on the Gaussian filtered image.

3.7.3 Gaussian Filtering

Gaussian filtering is mostly utilized to blur images and to remove noise. In one dimensional, the Gaussian function is written as:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (3.18)$$

where σ is the standard deviation with a mean of zero. Gaussian filtering is utilized in various research fields as it can define the probability distribution for noise or data and it is a smoothing operator. Note that with images, a two dimensional Gaussian function is considered. This is simply the product of two 1D Gaussian along the x and y axis direction (one for each axis) which is given by Equation 3.19.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.19)$$

3.7.4 Methodology of how Gaussian filtering works

The 2-D Gaussian distribution in the Gaussian filter works as a point spread function, which is then convolved with the image. Note that the kernel size is limited to 3σ as the distribution is very close to zero at 3σ (99%).

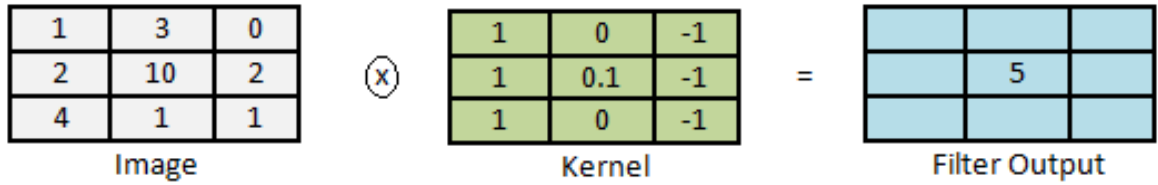


Figure 3.13: The process of filtering where the image is convolved with a kernel to give the filter output.

3.7.5 Image Denoising

In order to improve the thresholding, first the image is filtered so that grey values are more uniform. Filters used to this aim are called denoising filters, since their action accounts to reduce the intensity of the noise on the image. Several denoising filters try to average pixels together that are close to each other. One of the most common denoising filters is the Gaussian filtering. Zooming on a part of the image that should be uniform as seen in Figure 3.14, illustrates well the concept of noise. The image has random variations of gray levels that originate

from the imaging process. The origin of noise can be due to many sources, for example low photon-counting, electronic noise on the sensor or in our case imperfect deconvolution.

Figure 3.14 demonstrates a section of the original image before and after applying the gaussian filtering. It appears that the noise level has decreased and also spatially smoother.

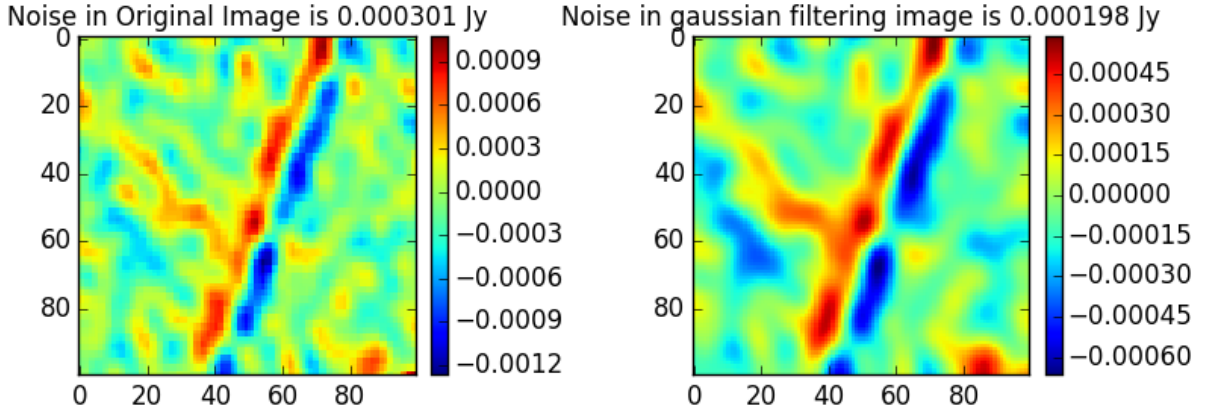


Figure 3.14: A region in the 2MASX J02581124-5243419 image where there is no source, is demonstrated in the left panel. Image denoising of 2MASX J02581124-5243419 using Gaussian filtering is shown in the right panel. The noise in the image has decreased after the application of Gaussian filtering on the image.

3.7.6 Applying Otsu Thresholding on the Gaussian filtered images.

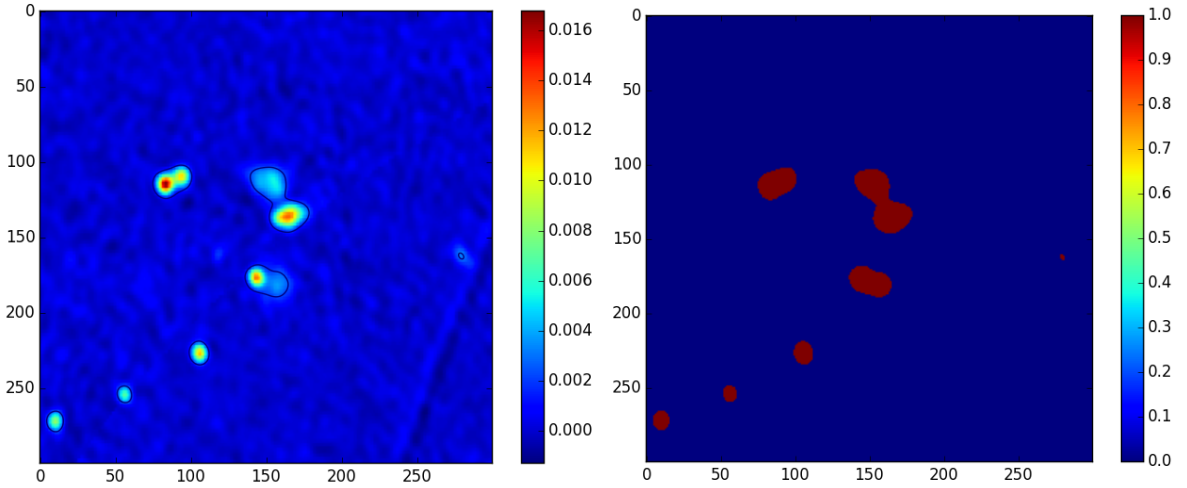


Figure 3.15: The left panel shows Otsu thresholding contours on Gaussian Filtered image 2MASX J02581124-5243419. The right panel illustrates the binary image after applying Otsu thresholding on the Gaussian filtered image. Both extended and point sources are extracted.

Instead of applying Otsu thresholding on the original image, it is implemented on the Gaussian filtered image. In the left panel of Figure 3.15, it is noticed that the Otsu thresholding performed a much better segmentation after applying the Gaussian filtering. The binary image after the

thresholding is shown in the right panel of Figure 3.15 where point and extended sources are assigned a pixel value of 1 and the background has zero values. It can be noted that the sources that were considered as two different sources by the other methods applied in the previous sections, has been successfully extracted as a single connected source with this method.

3.7.7 Extracting only extended sources.

If we use the denoising together with the thresholding approach, the result of the thresholding is not completely what we want to achieve since small point sources are also detected. Such defects of the segmentation can be amended, using the knowledge that no small point sources should exist.

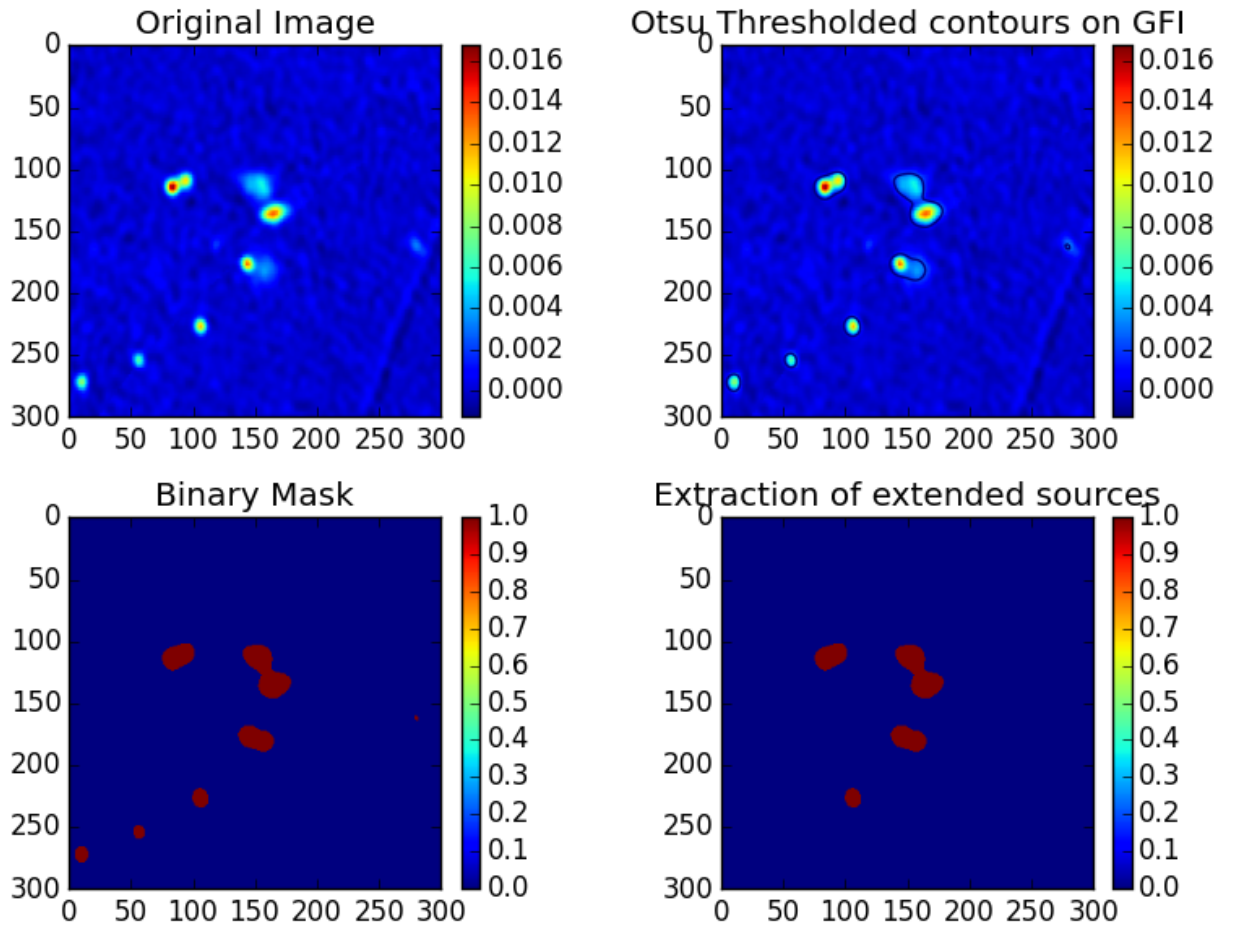


Figure 3.16: The process of extracting extended sources in 2MASX J02581124-5243419 image. A high-pass filtering algorithm is applied on the binary image (thresholded image), extracting only extended sources.

Therefore, a high-pass filtering algorithm is applied by assigning a minimal size for the extended sources. Therefore, all sources that are smaller than this size are classified as point sources and removed from the image. The minimal size is chosen according to the resolution, of the data R , that is, point sources have their angular size, $\theta < R$. It is seen in Figure ?? that 4

extended sources are detected in the 2MASX J02581124-5243419 image and 3 point sources are removed which agrees with the theoretical data for this image.

3.8 Remarks

It can be therefore concluded that extended sources have been successfully extracted from the images using the automated thresholding and filtering techniques compared to the LULU algorithm. Also, the Otsu thresholding and the filtering method is computationally efficient and feasible as it takes only 2 minutes to extract extended sources in the 151 images in the SUMSS data compared to the LULU algorithm which runs 8 hours to extract sources.

3.9 Summary

In this chapter, we have thoroughly described the concept of segmentation which is the allocation of every pixel in an image to one category or more that correspond to objects. The segmentation algorithms have been applied both directly to original images as well as after applying some filters. Firstly, thresholding has been applied manually and then an algorithm known as Otsu thresholding is implemented. We have come across different methods of extracting quantitative information from images and various details about the sources are obtained. We have come to the conclusion that the combination of an automated thresholding and filtering method for extracting extended sources from the SUMSS images is accurate, efficient and is not computationally expensive.

Chapter 4

4 Source Classification using Machine Learning techniques

This chapter provides the reader with the necessary image processing background and discusses the data that are used for machine learning. The main goal of this chapter is to perform classification of radio sources, that is, Point-Extended source classification and FRI-FRII source classification. Section 4.2 gives a brief overview on Shapelet theory and finally, Sections 4.3-4.6 describe the datasets used for classification and briefly illustrate the results obtained from the machine learning algorithms.

4.1 Introduction to Image processing

Exploring and processing of images is a fundamental aspect of scientific workflows of many astronomical analyses. Image processing is a technique that converts an image into a working scientific format. Various techniques and methods need to be combined for better interactivity, versatility and performance. As explained in the previous chapter, operations are performed on the images in such a way that an enhanced image or meaningful information can be extracted. Therefore, we can consider a digital image as a multi-dimensional array of numbers representing some arbitrary intensity, that is, numbers indicating channels of red, green, and blue at a particular location on a grid of pixels. Many areas in observational astronomy require high precision and accuracy image analysis, for instance the search for supernovae (Riess 1998, Perlmutter et al. 1995), microlensing (Mao 1999) and weak gravitational lensing (Bartelmann & Schneider 2008). Many sophisticated data analysis packages have been developed, for example FOCAS (Jarvis & Tyson 1981), SExtractor (Bertin & Arnouts 1996), wavelet analysis (Strack et al. 1998), Fourier analysis and many more. In this work, a method for image analysis based

on a Shapelet transformation is presented.

4.2 Shapelets Theory

Shapelets can be defined as a linear decomposition of an object into a series of localised basis functions with different shapes (Refregier 2008). The orthonormal set of 2D basis functions is chosen from Hermite polynomials weighted by a Gaussian, which is similar to perturbation about a circular Gaussian.

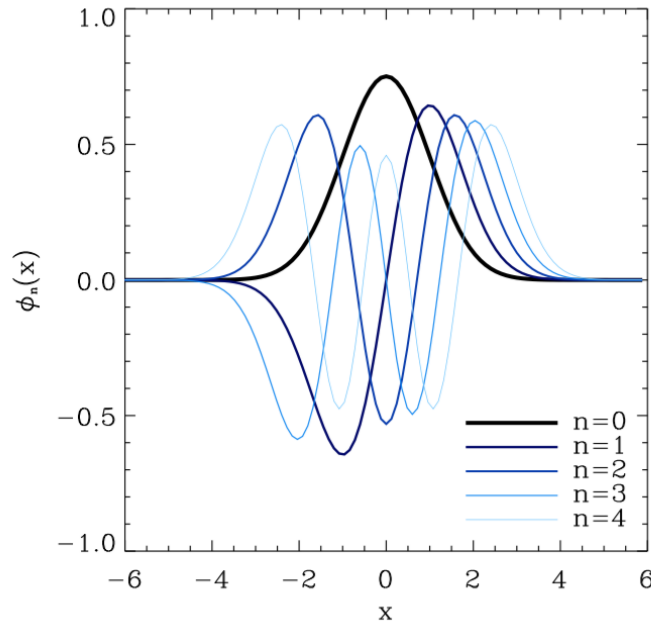


Figure 4.1: First few one dimensional basis functions $\phi_n(x)$. Figure from Refregier (2008).

In a similar way to Fourier or wavelet synthesis, a linear combination of these functions can be used to model any image. A remarkable property of shapelets is that they are invariant under Fourier transforms and hence give an analytical form for convolution (Refregier 2008). The method of shapelets actually decomposes an image into a series of compact disjoint objects of arbitrary shape which can be applied to astronomical images. This differs from the wavelet transform which decomposes an image into a sum of basis functions of different scale. The localised basis functions are chosen from a set of weighted hermite polynomials given in Equation 4.1. They are also the eigenstates of the 2-dimensional Quantum Harmonic Oscillator (QHO).

$$\phi_n(x) = \left[2^n \pi^{\frac{1}{2}}\right]^{-\frac{1}{2}} H_n(x) e^{-\frac{x^2}{2}} \quad (4.1)$$

where n is a non-negative integer and $H_n(x)$ is a hermite polynomial of order n which is

given by:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2} = \left(2x - \frac{d}{dx}\right)^n \cdot 1 \quad (4.2)$$

These functions are orthogonal and Figure 4.1 illustrates the first few one dimensional basis functions $\phi_n(x)$. In practice, an object can be described by the dimensional basis functions as given in Equation 4.3, where β is a characteristic scale, which is typically chosen to be close to the size of the object.

$$B_n(x; \beta) \equiv \beta^{\frac{1}{2}} \phi_n(\beta^{-1}x) \quad (4.3)$$

An object $f(x)$ can thus be decomposed as

$$f(x) = \sum f_n B(n) \quad (4.4)$$

where f_n are the shapelet coefficients given by

$$f_n = \int_{-\infty}^{\infty} f(x) B_n(x; \beta) dx \quad (4.5)$$

However, this can be extended for two dimensional objects which use two dimensional shapelets where radio images can be decomposed as in our investigation. The 2D shapelets functions can be constructed by taking the tensor product of two 1-dimensional basis functions. Thus, the dimensionless functions can be defined as

$$\phi_{\mathbf{n}}(\mathbf{x}) \equiv \phi_{n_1}(x_1) \phi_{n_2}(x_2) \quad (4.6)$$

where $\mathbf{x} = (x_1, x_2)$, $\mathbf{n} = (n_1, n_2)$ and the dimensional basis functions are given by

$$B_{\mathbf{n}}(\mathbf{x}; \beta) \equiv \beta^{-1} \phi_{\mathbf{n}}(\beta^{-1}\mathbf{x}) \quad (4.7)$$

The image of an object, that is, the 2-dimensional function $f(x)$ can thus be expressed as

$$f(x) = \sum_{n_1, n_2=0}^{\infty} f_{\mathbf{n}} B_{\mathbf{n}}(\mathbf{x}; \beta) \quad (4.8)$$

where the coefficients of the shapelets are given by

$$f_{\mathbf{n}} = \int \int f(\mathbf{x}) B_{\mathbf{n}}(\mathbf{x}; \beta) d^2\mathbf{x} \quad (4.9)$$

The first few two dimensional shapelet coefficients are illustrated in Figure 4.2 with $(n_1, n_2) =$ 5. The red and blue regions correspond to positive and negative values respectively.

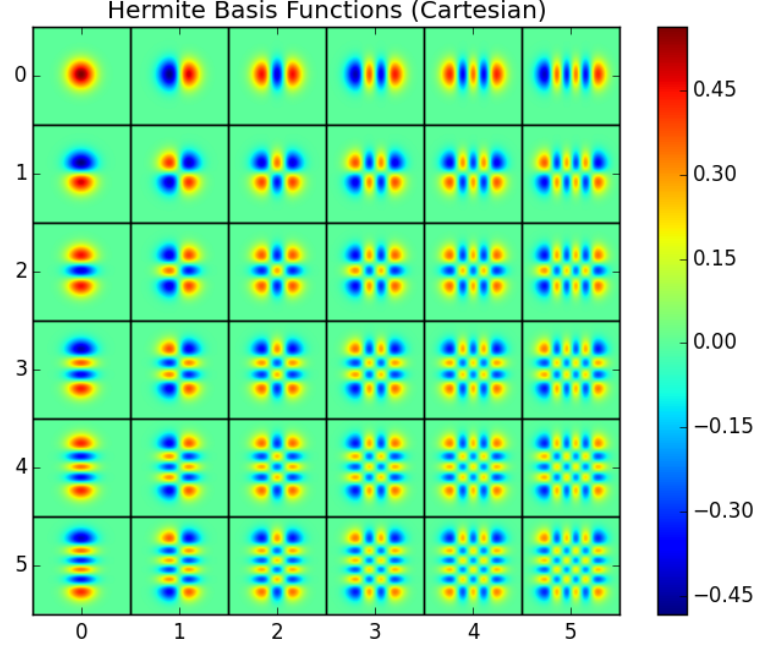


Figure 4.2: First few 2-dimensional Cartesian basis functions with $(n_1, n_2) = 5$.

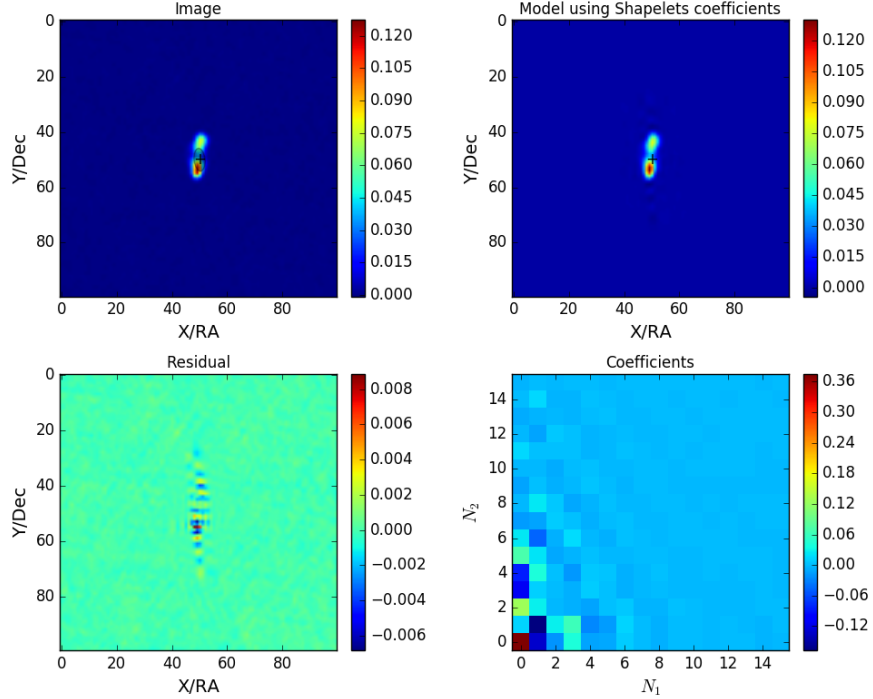


Figure 4.3: Application of shapelet decomposition on J154712+180410 radio image. The shapelet coefficients is shown in the bottom right panel and using these coefficients, a model image is reconstructed in the top right panel. The color bar indicates the flux in Jy.

An application of the shapelets method is implemented to decompose a radio image as illustrated in Figure 4.3. The J154712+180410 radio image is decomposed into its shapelets coefficients which is plotted on the bottom-right of Figure 4.3. Using the coefficients, a model is reconstructed and the residual image is simply the subtraction of the model from the original image.

The methodology utilized to obtain Figure 4.3, is described as follows:

- First, from the shapelets pseudocode module⁷, the module `plotImg.py` is used to plot the FITS image so that we have an idea where the source is located in the image. If sources are extracted from a catalogue, then the position/coordinate is already known where the source is located and how many pixels it covers. This will be a much faster way to select out sources from images. The function plots the image to allow us to determine the region where we want to implement the shapelet decomposition as seen in Figure 4.4.

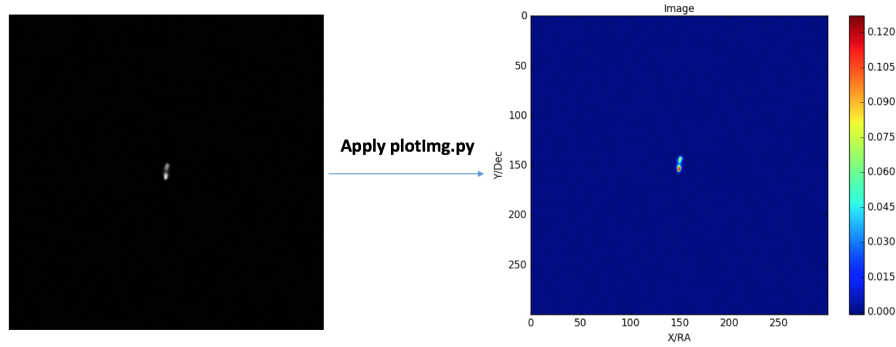


Figure 4.4: Using J154712+180410 radio source as input and generate the image using the module `plotImg`.

- Since the parameters β , ϕ , and the centroid position of the source are important to determine the shapelet coefficients, the script `fitShapelet.py` fits for all of these parameters. It is a generic script that allows one to specify which parameters to fit for, or which parameters to initialise to begin the fit. Since an initial guess of those parameters is not known, `fitShapelet.py` will try to pick ones which are reasonable enough to start the fitting.
- A set of parameters (β , ϕ , the coordinates of the decomposition) and an image as input are given as inputs to undertake the decomposition and then generate the shapelet coefficients. A file is generated where all the shapelet coefficients for the image are stored. The shapelet coefficients are further used to reconstruct a model of the original image. Then, a residual image is constructed by subtracting the original image with the model as shown in Figure 4.5.

⁷<https://github.com/griffinfoster/shapelets>

- Using the shapelets file where all coefficients are stored, this script (plotCoeffs.py) plots the coefficients and the model (reconstructed image using shapelet coefficients) as illustrated in Figure 4.6.

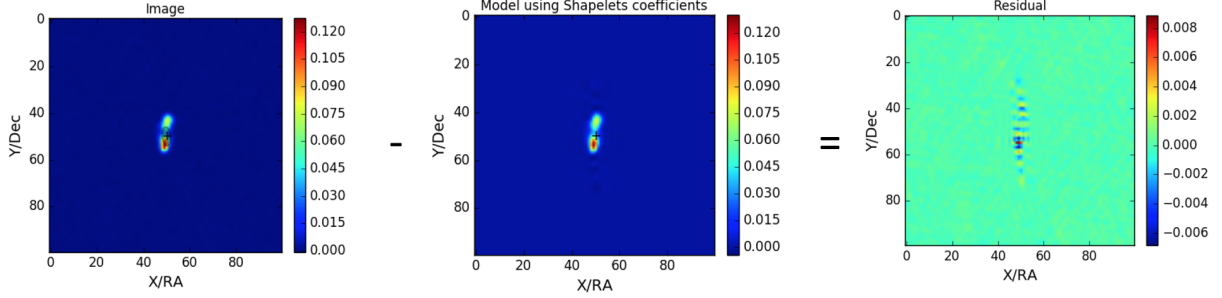


Figure 4.5: The residual image is obtained by performing a subtraction of the input image and the model image.

The shapelet coefficients can be used in various practical tools to compute the characteristics of the sources/objects for example the centroid, integrated flux, morphology, radius and orientation.

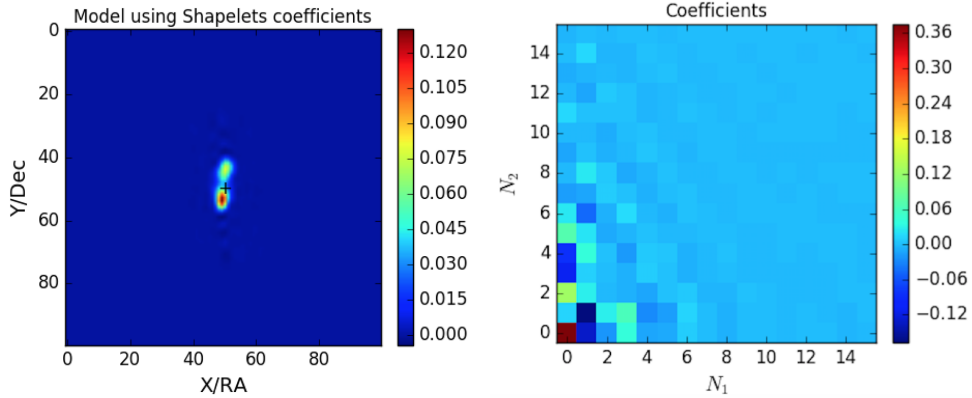


Figure 4.6: On the left, the plot of the model image reconstructed using the shapelet coefficients is illustrated and the shapelet coefficients are plotted on the right.

4.3 Sample selection for machine learning classification

A brief description of the data used to perform the classification is provided in Table 4.1.

In the dataset used, since the number of point sources and FRI radio sources are far less than that of the extended and FR II sources, we have used equal sizes for the classes to create a balanced data set. The FR II source being larger in number is only true for the data sample being used. The opposite is actually true in the universe, where powerful FR II sources are much rarer. For each catalogue, one example of each source is illustrated in Figure 4.7, that is,

point, extended, FRI and FR II sources respectively.

Table 4.1: Table summarizing the sample selection process.

| Types of Sources | Sample Size | Catalogues |
|------------------|-------------|---|
| Point Sources | 78 | Van Velzen et al. (2015) |
| Extended Sources | 78 | Van Velzen et al. (2015) |
| FRI Sources | 125 | Capetti et al. (2016), Becker et al. (1995), Condon et al. (1998) |
| FR II Sources | 125 | Capetti et al. (2017), Becker et al. (1995), Condon et al. (1998) |

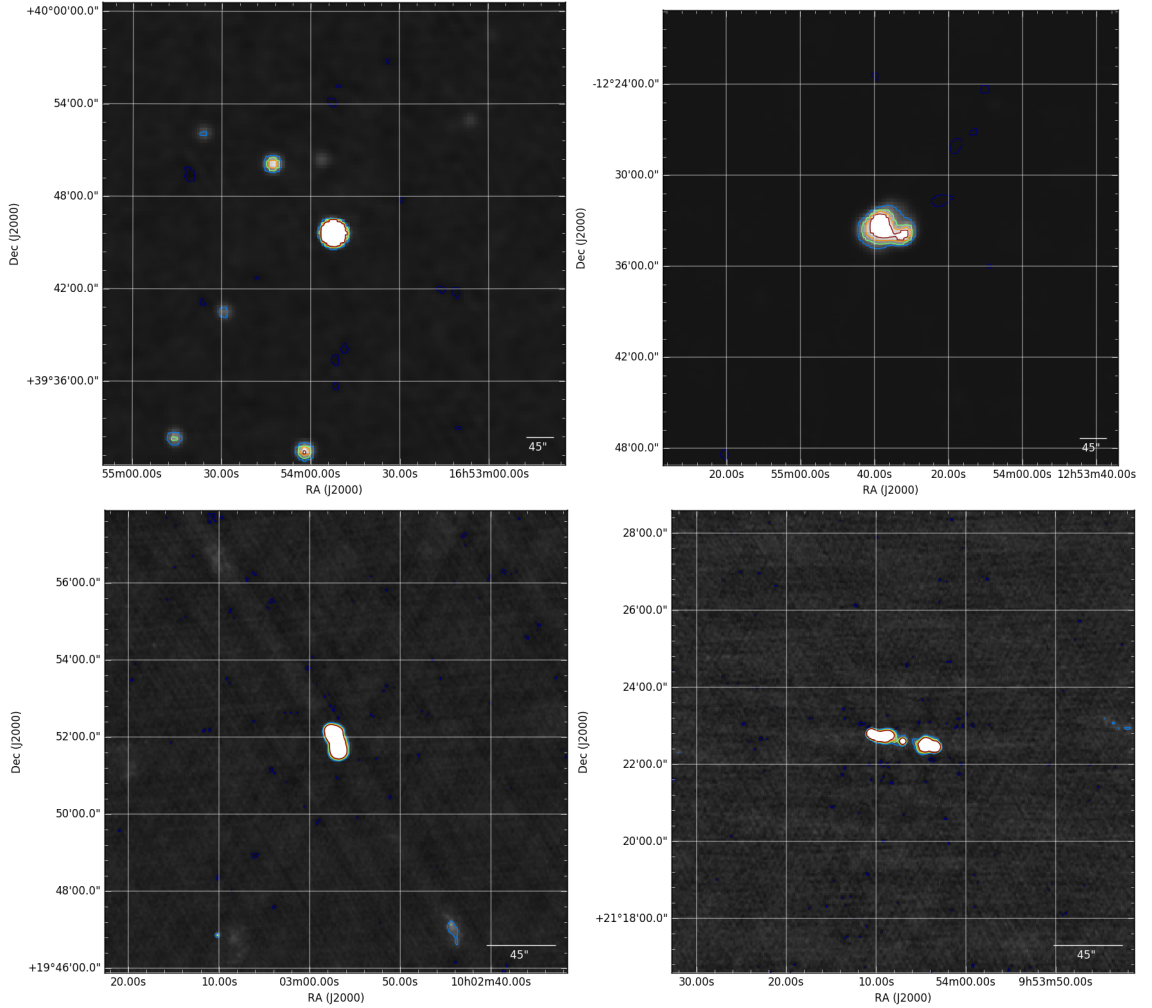


Figure 4.7: The four different classes of radio sources Point-like, Extended, FRI and FR II are shown respectively. Top Left: 2MASX J16535220+3945369 (point), Top Right: 2MASX J12543570-1234070 (extended), Bottom Left: SRC 22 (FRI), Bottom Right: SRC 14 (FR II).

4.4 Pipeline for Binary Classification using Machine Learning

This section gives a description of the methods used in the classification of the radio sources. The first step is to perform a classification of point-extended sources and secondly, a classification between FRI-FR II radio sources. Before any machine learning algorithms discussed

in Chapter 2 can be applied, there is a need to extract features from astronomical images. In this work, features from the astronomical images are obtained by performing a decomposition using the Shapelet transformation.

The framework for the classification is illustrated in Figure 4.8 and each step is explained in depth in later sections. Firstly, the astronomical images (point-like, extended, FRI and FRII sources) are decomposed into 256-coefficients using shapelet transformation. The shapelet coefficients (256) are the features of the images and a normalisation is applied to them. PCA is applied on these coefficients, thus reducing the dimensionality of the data by projecting them into a principal subspace. These coefficients are then visualised to know if there is a distinct separation between the classes. Then, the variance ratio is calculated to obtain the variances from the principal components to find how many dimensions of the data are more useful to better explain the entire dataset. We have noted that the first 40 coefficients could be better transformed by PCA, therefore the 40 shapelet coefficients act as inputs to the Machine learning algorithms (k Nearest Neighbours, Random Forest, Naive Bayes and Multi-Layer Perceptron). To avoid overfitting of the data, a stratified cross-validation and a randomized grid search for hyper-parameter optimization are used. Finally, the algorithms are evaluated and receiver operating characteristic (ROC) curves are plotted for each algorithm. In the next sections, a more detailed overview of each step will be discussed and illustrated.

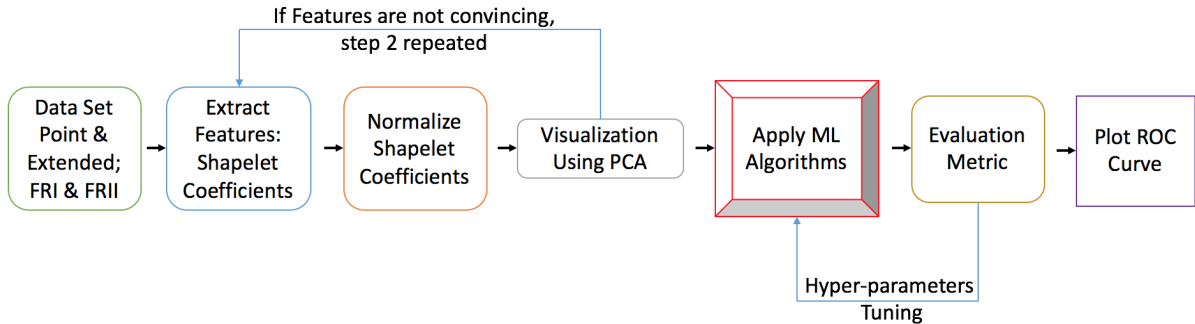


Figure 4.8: Illustration of the pipeline used for machine learning classification. Radio images are fed as input and shapelet decomposition is applied to extract features. The coefficients are analyzed and fed to the ML algorithms for classification.

4.4.1 Feature Extraction using the Shapelet Transform

Features of the sources are extracted by decomposing the astronomical images using the shapelet transform into 256 shapelet coefficients. For point and extended classification, the 156 sources are decomposed to generate vectors S_1, S_2, \dots, S_{156} where S is a vector having $f_{(n,n)}$ shapelet coefficients with a dimension of 16×16 as given in Equation 4.10.

$$\mathbf{S} = \begin{bmatrix} f_{(0,0)} & f_{(0,1)} & f_{(0,2)} & \cdots & f_{(0,15)} \\ f_{(1,0)} & f_{(1,1)} & f_{(1,2)} & \cdots & f_{(1,15)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{(15,0)} & f_{(15,1)} & f_{(15,2)} & \cdots & f_{(15,15)} \end{bmatrix} \quad (4.10)$$

Each source is then concatenated into a vector \mathbf{x} given in Equation 4.11.

$$\mathbf{x} = [f_{(0,0)} \cdots f_{(15,0)} f_{(0,1)} \cdots f_{(15,1)} \cdots \cdots f_{(15,0)} \cdots f_{(15,15)}] \quad (4.11)$$

The shapelet coefficients are complex valued, which means that each coefficient has amplitude and phase information. The phase provides information on the static angular position of the source and the amplitude is a function of the total flux. Then, a normalization of the vector \mathbf{x} is applied as given in Equation 4.12 where \mathbf{x}_N is the N^{th} coefficient. When normalization is performed, only the information about the shape of each source is stored. However, if two sources have the exact same shape and orientation with different amplitude once normalized, these two vectors will be equal.

$$\hat{\mathbf{x}} = \frac{|\mathbf{x}_N|}{\max(\mathbf{x})} \quad (4.12)$$

After the normalization, each source is then written in a single matrix, \mathbf{D}_{PE} for point-extended sources as written in Equation 4.13 and \mathbf{D}_{FRI-II} for FRI- FR II sources as given in Equation 4.14. We therefore end up with two matrices, \mathbf{D}_{PE} of dimension 156×256 where 156 is the total number of images for $P - E$ and \mathbf{D}_{FRI-II} of dimension 250×256 where 250 is the total number of images for FRI and FR II.

$$\mathbf{D}_{PE} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \vdots \\ \hat{\mathbf{x}}_{156} \end{bmatrix} \quad (4.13)$$

$$\mathbf{D}_{FRI-II} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \\ \vdots \\ \hat{\mathbf{x}}_{250} \end{bmatrix} \quad (4.14)$$

4.4.2 Data Visualization

Before a visualization of the 256 shapelet coefficients, we perform a rescaling of the coefficients such that the values are between 0 and 100. If the 256 coefficients are represented by a vector, then the rescaled value of the N^{th} coefficient is given by

$$\tilde{\mathbf{X}}_N = \frac{|\mathbf{X}_N|}{\max(\mathbf{X})} \times 100 \quad (4.15)$$

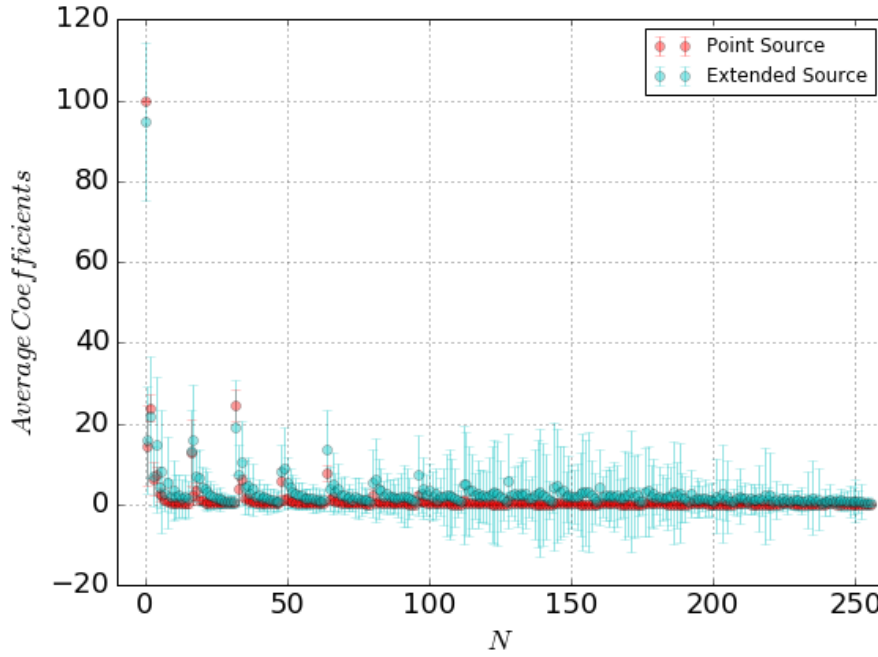


Figure 4.9: The average value of the coefficients for point-extended sources are plotted against the number of coefficients. A clear separation between point and extended sources is observed where point sources have a specific pattern.

Then, the average and the standard deviation of the rescaled coefficients $\tilde{\mathbf{X}}_N$ for the 78 point sources, 78 extended sources, 125 FRI sources and 125 FR II sources are computed separately. Figure 4.9 shows an illustration of the average and standard deviation of the 78 point-like and 78 extended sources against N (256 coefficients). It is observed that the coefficients of the point and extended sources have a definite pattern with their morphology and that the standard deviation for the extended sources are large, implying that there is a variation in the morphology of the sources compared to point sources. These procedures are repeated for the FRI and FR II sources are shown in Figure 4.10. However, we noticed for this particular class of radio sources, their coefficients have a similar pattern with no distinct separation.

To have deeper insight of the distribution of the data, the first three normalized coefficients as given in Equation 4.12 of the four different classes are plotted in Figure 4.11. Figure 4.11

illustrates the normalized coefficients of point and extended sources, showing a distinct region where the point sources (purple scatter points) reside. Figure 4.12 shows a 3D scatter plots for FRI-FRII sources using only the first three coefficients. Here, we noted that both FRIs and FRIIs show no distinct pattern that separates them.

Further, a dimensionality reduction is performed since we are dealing with 256 dimensions. We applied the PCA algorithm (see Section 2.3) which tries to find the directions of maximum variation in the data set. The aim of using a PCA is to identify the most important dimensions in the original feature space. Hence, the dimensionality of the data can drastically be reduced. In this case, the point-extended data consists of 156 samples and 256 features.

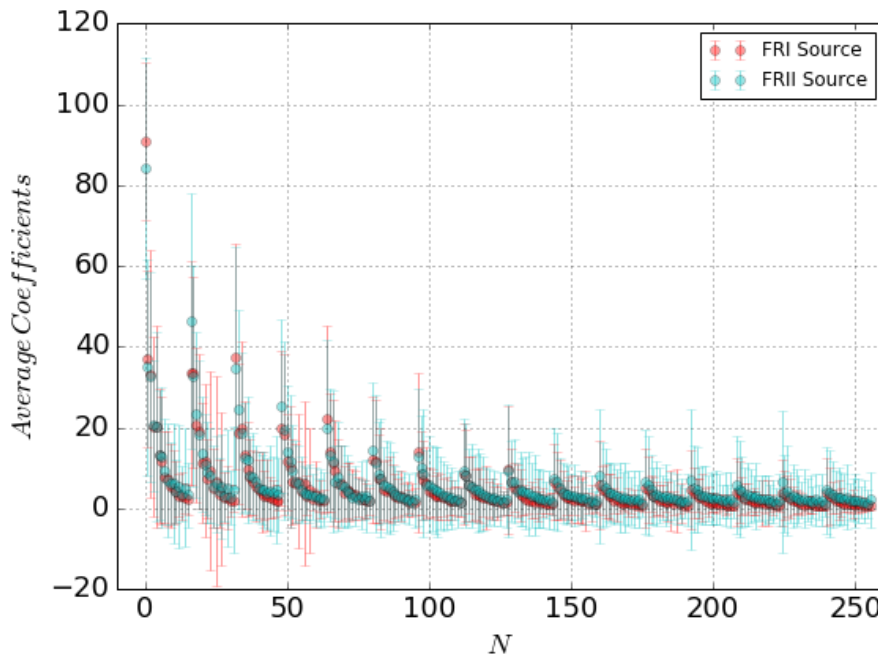


Figure 4.10: The average values of the coefficients for FRI-FRII sources are plotted against the number of coefficients. Similar patterns for FRI & FRII sources are seen with no clear separation between the sources.

A way to know how much information we retain when performing PCA is to look at the explained variance ratio of the principal component. We have noted that with 40 components, the data can be explained with 99.8% of the full variance. The same procedure has been implemented for FRI & FRII datasets where the latter can be explained with 95.9% of the variance with 40 coefficients. Once convinced that we retain sufficient variance in the dataset, we utilize the 40 coefficients in the four different machine learning algorithms: k Nearest Neighbours, Multi-layer Perceptron, Random Forest and Naive Bayes classifiers. The labels are generated as a 1Ds array with 1s and 0s, where 0s are for point sources and 1s are for extended sources while for FRI & FRII classification, 0s are for FRI and 1s are for FRII sources.

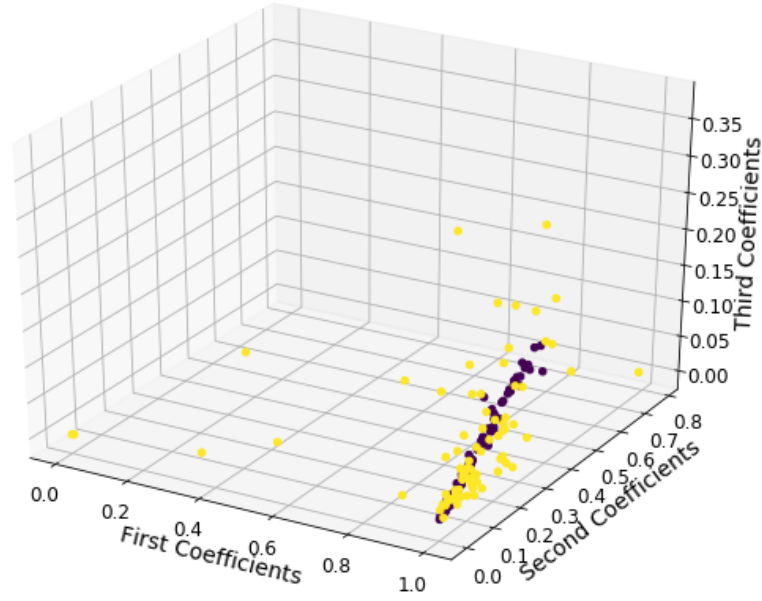


Figure 4.11: The first three normalized shapelet coefficients are plotted for point and extended sources. The purple scatter points represent the point sources while the yellow scatter points illustrate the extended sources. It is observed that the coefficients of the point sources lie in a specific region well separated from the coefficients of the extended sources.

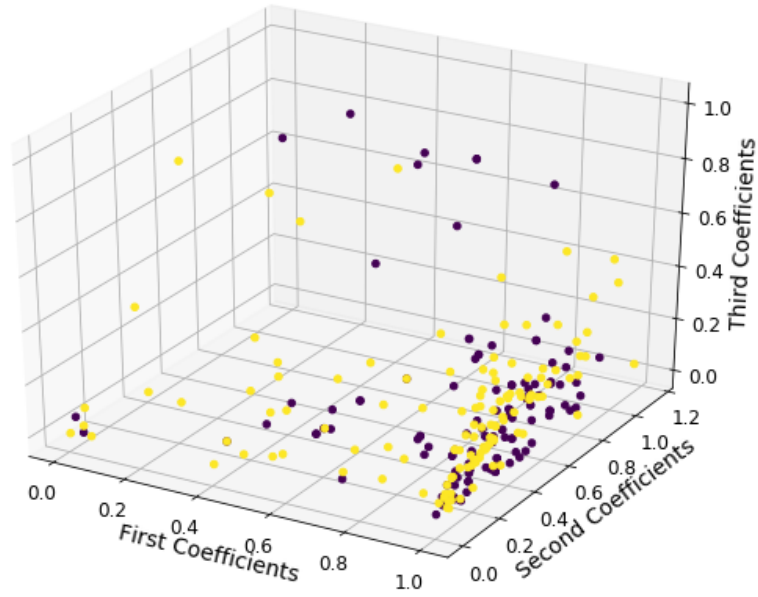


Figure 4.12: The first three normalized shapelet coefficients are plotted for FRI and FRII sources. The purple scatter points represent the FRI sources while the yellow scatter points illustrate the FRII sources. It is observed that the coefficients for the FRI and FRII are both scattered, with no distinct separation between each class.

4.5 Application of Machine Learning

In this section, the implementation, training and testing of each of the four algorithms that we implemented for our classification are discussed. Also, the performance for the different classifiers obtained from the final testing is given. The theoretical background for the ML al-

gorithms required for this section is discussed in Chapter 2. The implementation of the four different classification algorithms was done with scikit-learn (Pedregosa et al. 2011) (version 0.18.1) which is an open source machine learning library for Python⁸ (Python 3.5.3 is used in our case).

The evaluation of the different classifiers is done by splitting the whole data set into two portions known as train set and validate/test set. The training data is used to train the model while the test data is used to evaluate the performance of the trained model. In our study, we used stratified K-fold cross validation with 10 folds. A more detailed overview of K-fold cross validation, hyper-parameter optimization and the evaluation metrics for the ML algorithms, is given in the following sections.

4.5.1 K-Fold Cross-Validation

The major problem when using machine learning algorithms is the overoptimistic result. Overfitting mostly occurs when training and evaluation is performed on the same data. Therefore, to avoid such a problem, the algorithm must be tested on a new data which would output a good estimate of its performance accuracy. A statistical method often used to evaluate and compare ML algorithms is to split the dataset into train and test sets. The training set is utilized to train the algorithm and the model is validated using the test set. A method known as cross-validation, ensures that the train and test samples are successively crossing-over in such a way that each data point has the opportunity to be validated against each other (Refaeilzadeh et al. 2009).

For the case of K -fold cross-validation, firstly the data set is split into K equally or nearly equal sized folds. K iterations are performed such that for each iteration, a different set of the data is kept-out for validation and the rest $K - 1$ sets are utilized for training. In addition, a stratification of the data is applied such that for each set of the $K - fold$, the data are arranged to ensure each fold preserves the percentage of samples for each class. Each fold consists mostly the same amount of predictor labels as the entire dataset. In K -fold cross-validation, the dataset D is split randomly into K mutually exclusive subsets (D_1, D_2, \dots, D_K) of nearly the same size. For K times, the classifier algorithm is trained and tested. For each time $t \in \{1, 2, \dots, K\}$, the algorithm is trained on $K - 1$ folds and one fold D_t is used for performing validation. The overall accuracy of algorithm when using cross-validation is simply the average of each K accuracy measures given in Equation 4.16.

⁸<https://www.python.org/>

$$A_{cv} = \frac{1}{K} \sum_{i=1}^K A_i \quad (4.16)$$

where A_{cv} is the cross-validation accuracy, A is the accuracy measure for each fold, K is the number of fold (Dursun 2009).

4.5.2 Hyper-parameter optimization and Evaluation Metric

The hyper-parameter optimization is considered as a problem for the various classifiers. The most widely used techniques for hyper-parameter optimization are the grid search and manual search. Grid search is very computationally expensive as it tries all combinations of the hyper-parameters given manually and will output the report that leads to the highest accuracy. However, in our study, we applied a randomized search that iterates through the pre-specified hyper-parameters a number of times for some distribution and find the optimum that output the best accuracy for the classifier.

In addition, it is difficult to decide which metrics are the most suitable to evaluate the performance of algorithms and often the predictive accuracy is used. This can be misleading in the case where we have unbalanced datasets (Chawla et al. 2002). The evaluation of the performance of Machine Learning algorithms requires a certain level of trade-off between the number of true/false positives/negatives rate, between recall and precision. In addition, another measure is the Receiver Operating Characteristic (ROC) curve which represents graphically the trade-off between false negative and false positive rates.

The performance of any machine learning algorithm is computed using measures such as the accuracy, the precision, the recall, the F1 score, sensitivity and specificity (Chao et al. 2004). They are defined in terms of True Positive (T_P) Rate, False Positive (F_P) Rate, True Negative (T_N) Rate and False Negative (F_N) Rate. The sensitivity metrics is defined as the true positive rate or positive class accuracy, while specificity is referred to as true negative rate or negative class accuracy (Danjuma 2015).

- Accuracy Measure: It compares how close a new test data is to a value predicted (Ciosa & Moore 2002) and it is given in Equation 4.17. In cases where the number of positive and negative classes are largely different, accuracy will lead to misleading results. Our dataset consists of balanced classes, therefore it is reasonable to apply the accuracy as an initial measure.

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \times 100\% \quad (4.17)$$

- Sensitivity Measure: Equation 4.18 is also known as the true positive rate or recall that measures the proportion of actual positives which are correctly identified by the model.

$$\text{Sensitivity/Recall} = \frac{T_P}{T_P + F_N} \quad (4.18)$$

- Specificity Measure: Equation 4.19 also known as True Negative rate that measures the ability to identify negative results.

$$\text{Specificity} = \frac{T_N}{T_N + F_P} \quad (4.19)$$

- Precision: It is a measure of retrieved instances that are accurate which is given in Equation 4.20.

$$\text{Precision} = \frac{T_P}{T_P + F_P} \quad (4.20)$$

- F1-score: It is expressed in terms of the precision and the recall given in Equation 4.21.

$$F1 - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.21)$$

The T_P , F_P , T_N , and F_N can be visualized by a confusion matrix as illustrated in Table 4.2, where samples in the predicted class are given in each column and samples in an actual class correspond in each row. In this case, from Table 4.2, the true positives (T_P) are point sources that were correctly classified as point, false positives F_P correspond to extended sources wrongly classified as point. And in a similar way, false negatives F_N and true negatives T_N can be explained. For the point & extended classification problem, point sources correspond to positive class and extended sources correspond to negative class. And for FRI & FRII classification, FRI sources is assigned to the positive class and FRII sources is assigned the negative class.

Table 4.2: Confusion matrix implemented for our specific problem, where the positive class corresponds to point and FRI sources and the negative class to extended and FRII sources for the two classification schemes. True/false positives/negatives are represented as T_P , F_P , T_N , and F_N respectively.

| | Extended | | Point | | | FRII | FRI |
|--------------|----------|-------|-------|-----------------|------|-------|-------|
| | Extended | T_N | F_P | Actual Class | FRII | T_N | F_P |
| Actual Class | Point | F_N | T_P | Predicted Class | FRI | F_N | T_P |
| | | | | | | | |

4.5.3 Receiver Operating Characteristics (ROC)

In a more systematic way, trade-offs between true positive T_P and false positive F_P rates can be studied as most classifiers output the probability of a source being in a specific class. This

is done by using the Receiver Operating Characteristic (ROC) curve. This a graphical representation of the True Positive Rate against the False Positive Rate at various threshold values (that is, adjusting some threshold value that control the number of examples labelled true or false) and also the performance of the binary classification is shown using this curve. The False Positive Rate (FPR) is computed as given in Equation 4.22. The Area under the Curve (AUC) is the most preferred performance metric in the machine learning community for the ROC curve. The AUC is most accepted because the larger the AUC the better the classifier is. In addition, the AUC interprets nicely the probability that the classifier will correctly classify a randomly chosen positive instance above a randomly chosen negative one (Ian et al. 2011).

$$FPR = \frac{F_p}{F_p + T_N} \quad (4.22)$$

4.6 Results and Analysis

For both classifications (point-extended and FRI-FRII), the accuracy serves as an adequate metric since our datasets are balanced (similar sizes). Figure 4.13 shows the ROC curves for the four classifiers implemented on the test set with their corresponding AUC for point-extended classification while Figure 4.14 shows the ROC curves for FRI-FRII classification.

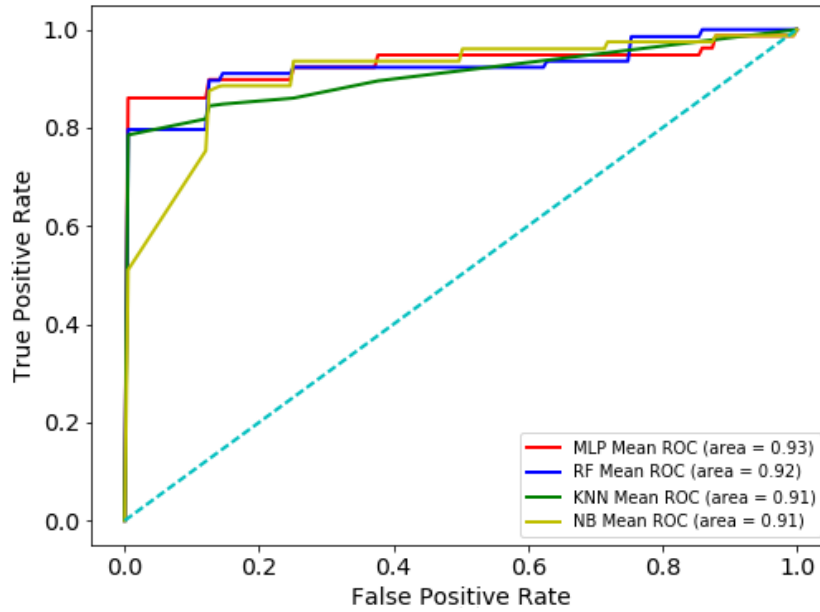


Figure 4.13: The ROC curve and the Area Under Curve (AUC) for each of the four classifiers for Point-Extended classification is shown. Next to each classifiers name, the AUC statistic is indicated in parentheses.

The main results from the machine learning techniques for source classification are given in Table 4.3, Table 4.4, Figure 4.13 and Figure 4.14. Table 4.3 and Table 4.4 provide a summary

of the performance of the different classifiers and they are listed from best performing to worst performing based on the value of the accuracy. From Table 4.3, it is observed that the Multi-Layer Perceptron (MLP) performs best in terms of all the metrics including the AUC (see Figure 4.13) for point-extended classification followed by Random forest (RF), k -Nearest Neighbours (k NN) and naive Bayes (NB). From Table 4.4, for FRI-FRII classification, RF is the best classifier for all the metrics. In addition, from Table 4.3, we note that for all the validation data of the point-extended classification, the MLP model shows excellent average precision (96%), average recall (84%) with an F1-score of 88%. For FRI and FRII classification, the random forest classifier is the best performing model and shows an average precision of 74%, average recall of 82% with an F1-score of 77% which implies that the models were not that successful to accurately distinguish between FRI and FRII sources.

Table 4.3: A summary of the performance results of the various classifiers, ordered from best-performing to worst-performing for classification of Point-Extended sources. The best result for each performance metric is indicated in bold red. The true labels are on the left side of the confusion matrices where Ext stands for Extended sources, and the predicted labels are at the top.

| Machine Learning Techniques | Precision | Recall | F1-Score | Accuracy | AUC | Confusion Matrix | | |
|---|-----------|-----------|-----------|-----------|------|------------------|-----|-------|
| Point and Extended Source Classification | | | | | | | | |
| Multi-Layer Perceptron (MLP) | 0.96±0.06 | 0.84±0.18 | 0.88±0.11 | 0.89±0.08 | 0.93 | | Ext | Point |
| | | | | | | Ext | 75 | 3 |
| | | | | | | Point | 14 | 64 |
| | | | | | | | | |
| Random Forest (RF) | 0.92±0.09 | 0.79±0.20 | 0.88±0.08 | 0.87±0.10 | 0.92 | | Ext | Point |
| | | | | | | Ext | 73 | 5 |
| | | | | | | Point | 16 | 62 |
| | | | | | | | | |
| <i>k</i> -Nearest Neighbours (<i>k</i> NN) | 0.96±0.07 | 0.71±0.23 | 0.81±0.18 | 0.83±0.11 | 0.91 | | Ext | Point |
| | | | | | | Ext | 76 | 2 |
| | | | | | | Point | 24 | 54 |
| | | | | | | | | |
| Naive Bayes (NB) | 0.90±0.13 | 0.56±0.21 | 0.68±0.19 | 0.75±0.13 | 0.91 | | Ext | Point |
| | | | | | | Ext | 74 | 4 |
| | | | | | | Point | 35 | 43 |
| | | | | | | | | |

In conclusion, we observed that the feature extraction using the shapelet transform per-

formed very well using machine learning techniques for point-extended classification. While for the case of FRI-FRII classification, the shapelet transform for feature extraction did not encapsulate the distinct features of the sources accurately. Thus, machine learning algorithms were not successful in classifying well the FRI and FRII sources. Therefore, in the next chapter we will employ a deep learning technique for the classification of FRI and FRII sources which is advantageous, as it learns directly from the images rather from features.

Table 4.4: A summary of the performance results of the various classifiers, ordered from best-performing to worst-performing for classification of FRI-FRII sources. The best result for each performance metric is indicated in bold red. The true labels are on the left side of the confusion matrices, and the predicted labels are at the top.

| Machine Learning Techniques | Precision | Recall | F1-Score | Accuracy | AUC | Confusion Matrix | | |
|---|-----------|-----------|-----------|-----------|------|------------------|------|-----|
| FRI and FRII Source Classification | | | | | | | | |
| Random Forest (RF) | 0.74±0.10 | 0.82±0.09 | 0.77±0.06 | 0.75±0.08 | 0.74 | | FRII | FRI |
| | | | | | | FRII | 85 | 40 |
| | | | | | | FRI | 23 | 102 |
| Multi-Layer Perceptron (MLP) | 0.69±0.09 | 0.67±0.13 | 0.68±0.10 | 0.68±0.09 | 0.66 | | FRII | FRI |
| | | | | | | FRII | 87 | 38 |
| | | | | | | FRI | 41 | 84 |
| <i>k</i> -Nearest Neighbours (<i>k</i> NN) | 0.67±0.09 | 0.62±0.11 | 0.64±0.09 | 0.65±0.07 | 0.69 | | FRII | FRI |
| | | | | | | FRII | 85 | 40 |
| | | | | | | FRI | 47 | 78 |
| Naive Bayes (NB) | 0.72±0.26 | 0.24±0.14 | 0.34±0.16 | 0.56±0.10 | 0.61 | | FRII | FRI |
| | | | | | | FRII | 109 | 16 |
| | | | | | | FRI | 95 | 30 |

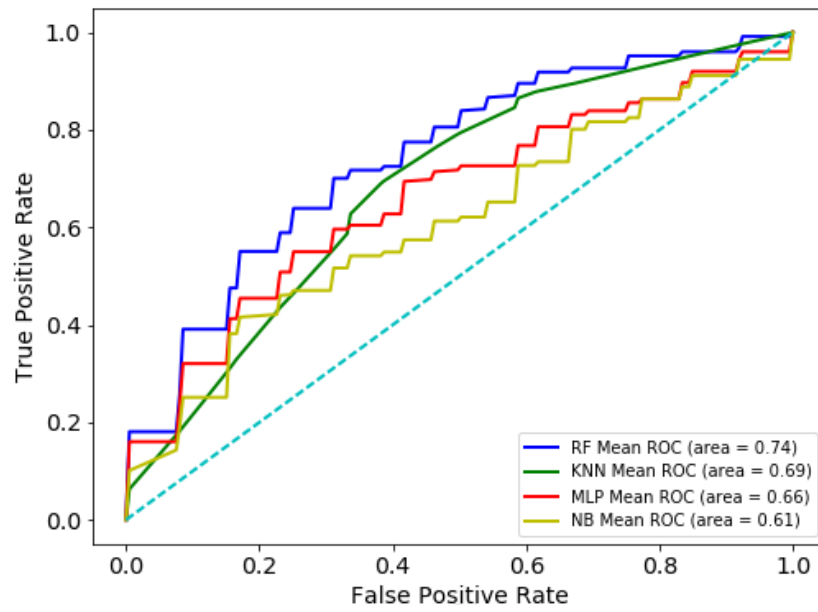


Figure 4.14: The ROC curve and the Area Under Curve (AUC) for each of the four classifiers for FRI-FRII classification is illustrated. Next to each classifiers name, the AUC statistic is indicated in parentheses.

Chapter 5

5 Source Classification using Deep Learning

This chapter provides the reader with some deep learning background and discusses the data that is used for classification purposes. The main goal of this chapter is to improve the classification of FRI-FRII radio sources. Section 5.7 gives a brief overview on convolutional neural network theory. Finally we discuss the dataset used and present the results from the deep learning algorithms.

5.1 Introduction

In this chapter, deep learning techniques are applied to perform the classification of AGN-powered radio galaxies, more specifically the FRI and FRII sources. Section 2.4 provides an overview of the application of machine learning techniques in astronomy. However, we have observed that machine learning algorithms for instance k NN, MLP, RF, and NB require features that are extracted from the images where those features represent specific and unique physical properties of the sources. The quality of the features used is an important determinant in the success of the ML algorithms. This technique is known as shallow learning (Chen 1995). Therefore, feature extraction is a fundamental and difficult process in the machine learning domain.

Deep learning is a subset of machine learning in which the algorithms learn the features directly from the data. This becomes important in cases where features extracted do not completely capture the physical characteristics of the raw data. With the accessibility and performance of modern day computing resources, deep learning has seen significant development and rapid deployment to numerous applications. LeCun et al. (2011) demonstrated that Deep

Neural Networks (DNNs) perform better on certain problem domains than shallow learning with very high accuracy. DNNs have been applied in various fields such as object recognition (Szegedy et al. 2013), speech recognition (Ossama et al. 2014) and image captioning (Vinyals et al. 2015). After considering these advancements, DNN is a fundamental tool that can be applied towards the classification of FRI and FR II radio sources. In this work, the classification is based on the morphology of the sources only.

5.2 Sample Selection for Deep Neural Network

This section describes the data used to perform the classification. A combination of different samples of NVSS and FIRST galaxies sources with well-resolved images have been selected to make two separate data sets of FRI and FR II. Since the support for FRI and FR II sources is quite low, the FRICAT and FR II CAT catalogues have also been included (see Section 1.6.2). Considering the free availability and well-resolved images, we have 171 FRI images and 646 FR II images. For the deep learning algorithm, we have applied the following criteria for distinguishing between FRI and FR II sources: sources with two distinct hotspots were classified as FR II category and FRI galaxies were sources having one hotspot close to the core. Also, images with strong artefacts and multiple sources are all excluded in the final samples. After applying these criteria, we are left with 96 FR Is and 369 FR IIs. Final samples used for the deep learning algorithms are summarized in Table 5.1.

Table 5.1: Table summarizing the sample selection process for the DL algorithms used in this study.

| Types of Sources | Initial Dataset | Final Dataset | Catalogues |
|------------------|-----------------|---------------|---|
| FRI Sources | 171 | 96 | Capetti et al. (2016), Becker et al. (1995), Condon et al. (1998) |
| FR II Sources | 646 | 369 | Capetti et al. (2017), Becker et al. (1995), Condon et al. (1998) |

5.3 Preprocessing images and data augmentation

When training a neural network, a recurrent problem faced is that not enough data is available to maximize the generalization capability of DNNs. Many techniques have attempted to resolve this issue, like data augmentation, dropout and transfer learning (Yosinski et al. 2014). Data augmentation can be defined as a process of supplementing a dataset with similar data that is constructed from features and information in that dataset. In this work, we demonstrate

three methods for data augmentation to increase the accuracy and reduce overfitting of the neural networks. Training a deep neural network effectively requires a large training dataset of labelled examples. When training a deep neural network on a small dataset, the network will likely overfit, i.e. it will memorize the examples that it has seen instead of learning the most salient features that will allow it to generalize in classifying new unseen examples at test time.

Other techniques such as using a different regularization approach (Khan 2001) have been proposed to solve this problem. Recently, the different regularization techniques such as the dropout techniques (Srivastava et al. 2014) and batch normalization (Ioffe & Szegedy 2015a) have successfully been used for deep neural networks to avoid overfitting on most data. Data augmentation performs a type of regularization that extracts more important information from the dataset and passes it through the network, thus reducing the chance of overfitting. The augmentation process can be classified into two different types: supervised and unsupervised augmentation.

Supervised augmentation involves the mixture of different samples with the same label in order to generate a new sample with the same label and the reconstructed sample should have the same features as a valid data sample. This kind of augmentation involves the use of the labels of the data, thus known as supervised. For unsupervised augmentation, the data expansion is processed without considering the label of the data. This involves adding various kinds of noise, rotation and flipping of the data.

In Simard et al. (2003) and Chatfield et al. (2014), an in-depth description of manual augmentation techniques (rotation, flipping and adding different kinds of noise to the data samples) is given and they also suggest a catalogue of data augmentation techniques. Srivastava et al. (2014) implemented the dropout technique which is the process of removing a unit (or artificial neuron) from the Artificial Neural Network (ANN) to reduce overfitting. In another study, Konda et al. (2015) used dropout as an augmentation method by projecting the noise of the dropout within a network back into the input space. Another possible consideration for data augmentation, which was used in this study, is to train an adversarial network to generate synthetic images that resemble the original images (Goodfellow et al. 2014). In addition, transfer learning is another method to increase the neural network generalization capacity. It involves acquiring knowledge from one network and transferred it to another (Shrivastava et al. 2016).

Since the Convolution Neural Network (CNN) will later be applied to perform the classification of FRI and FRII, the use of data augmentation in deep learning is essential. Therefore,

since our dataset is mainly composed of a small sample of FRI and FRII, three techniques have been implemented for data augmentation which includes (i) reconstruction of an image using a sampling approach from shapelet coefficients, (ii) application of rotation and flipping and finally (iii) the implementation of a Deep Convolutional Generative Adversarial Network (DCGAN).

5.4 Method 1: Reconstructing model images using shapelet coefficients

The aim is to reconstruct model images from the shapelet coefficients after decomposing the original input image. This is achieved by implementing the sampling method from a probability distribution.

5.4.1 Sampling from a probability distribution

A sampling distribution is the probability distribution of a given statistic based on a random sample. The normal (or Gaussian) distribution is a very common continuous probability distribution that is often used in probabilistic theory. The normal distribution refers to a family of continuous probability distributions described by the normal equation and it is useful because of the central limit theorem (CLT). The CLT states that if a large number of identically distributed random variables are added, the distribution of the sum will be approximately normal under certain conditions.

5.4.2 Reconstructing model images through sampling from 256 shapelet space

A random variable \mathbf{x} is said to be normally distributed with mean μ and standard deviation σ if its probability distribution is given by

$$f(\mathbf{x}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5.1)$$

If we consider $\mathbf{x} \sim \mathcal{N}(0, 1)$, that is, $\mu = 0$ and $\sigma = 1$, Equation 5.1 becomes

$$f(\mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (5.2)$$

In Chapter 4, the shapelet theory has been discussed and we have seen that an image is decomposed to generate a vector \mathbf{S} having $f_{(n,n)}$ shapelet coefficients with a dimension of 16×16 . Then, a model is constructed from these coefficients.

$$\mathbf{S} = \begin{bmatrix} f_{(0,0)} & f_{(0,1)} & f_{(0,2)} & \cdots & f_{(0,15)} \\ f_{(1,0)} & f_{(1,1)} & f_{(1,2)} & \cdots & f_{(1,15)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{(15,0)} & f_{(15,1)} & f_{(15,2)} & \cdots & f_{(15,15)} \end{bmatrix} \quad (5.3)$$

The vector \mathbf{S} is further used to generate many pseudo-images from the original image. This is achieved by sampling through the $f_{(n,n)}$ shapelet coefficients. If \mathbf{S} is the vector of 256 shapelet coefficients, new coefficients, \mathbf{S}_{new} are sampled from the standardized normal distribution with mean 0 and variance 1 as given in Equation 5.4 where each coefficient f is perturbed by $\sigma \times f \times \mathcal{N}(0, 1)$ and σ is the percentage that we want to vary the coefficient around the central value f .

$$\mathbf{S}_{new} = \mathbf{S} [1 + (\sigma \times \mathcal{N}(0, 1))] \quad (5.4)$$

Figure 5.1 shows the sampling distribution for coefficient $f_{(0,0)}$. The red dashed line is the value of $f_{(0,0)}$ (true coefficient value) and the blue dashed line is the new coefficient sampled from the distribution.

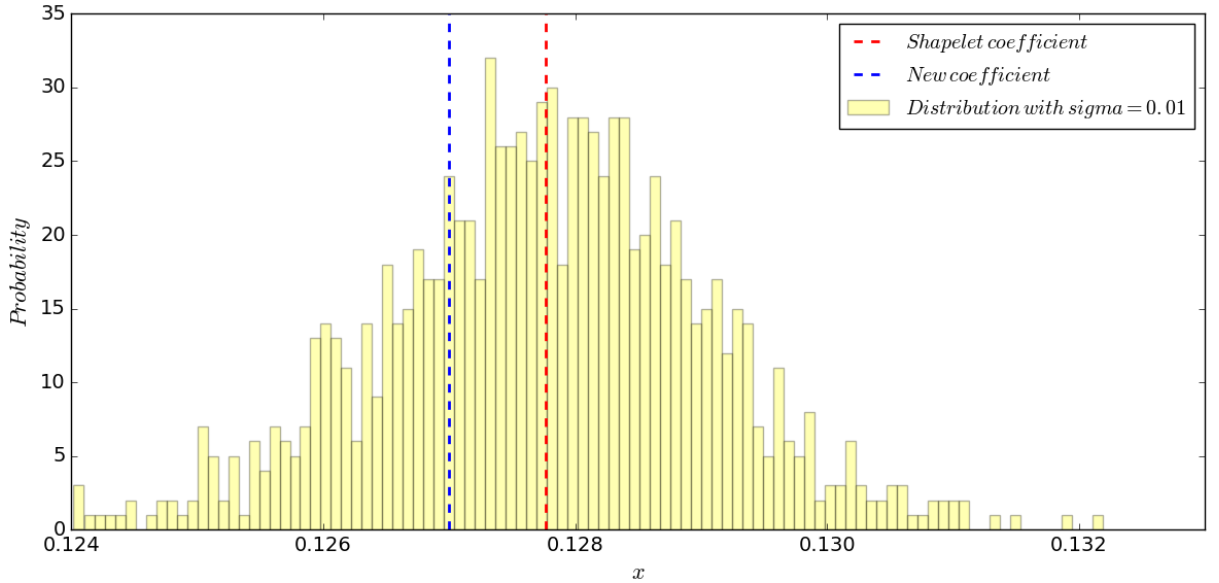


Figure 5.1: Sampling new coefficient from the distribution. The true shapelet coefficient is represented by the red dashed line and the blue dashed line is an example of a new sampled coefficient. x is the distribution of a single coefficient f_{00} obtained from \mathbf{S}_{new} following Equation 5.4. The y-axis is the probability/frequency/likelihood of each particular points, x . We are sampling from this probability distribution to obtain a new coefficient.

Finally, new coefficients are generated by sampling from those shapelet coefficients as given in Equation 5.4 and new model images are reconstructed by varying σ . Two different image classes are used to demonstrate the process of image reconstruction using shapelets. These

example images, namely MRC0007-287 and MRC0020-253, are taken from the two classes FRI and FRII, respectively. σ is varied and different results of the reconstructed model are shown for the FRI and FRII sources.

5.4.3 Reconstructing Image with different σ s

The MRC0007-287 and MRC0020-253 images are decomposed using shapelet transform and their model images are reconstructed as shown in Figure 5.2 and Figure 5.3. It is observed in Figure 5.4, mostly the same model is reconstructed with $\sigma = 0.1$ when compared to its original image in Figure 5.2 and only a slight change in the residual image was observed. The residual images are not shown for all the σ s variation but it was observed that the pattern in the residual image kept on changing for the different σ s. Also, in Figure 5.4, it is noticed that as σ is varied, different models are constructed. As σ is increased, a completely new model is generated and the model becomes noisy. The same procedures are repeated for the image MRC0020-253. It is observed that for $\sigma > 0.2$, the reconstructed model images start to deviate significantly from the original image.

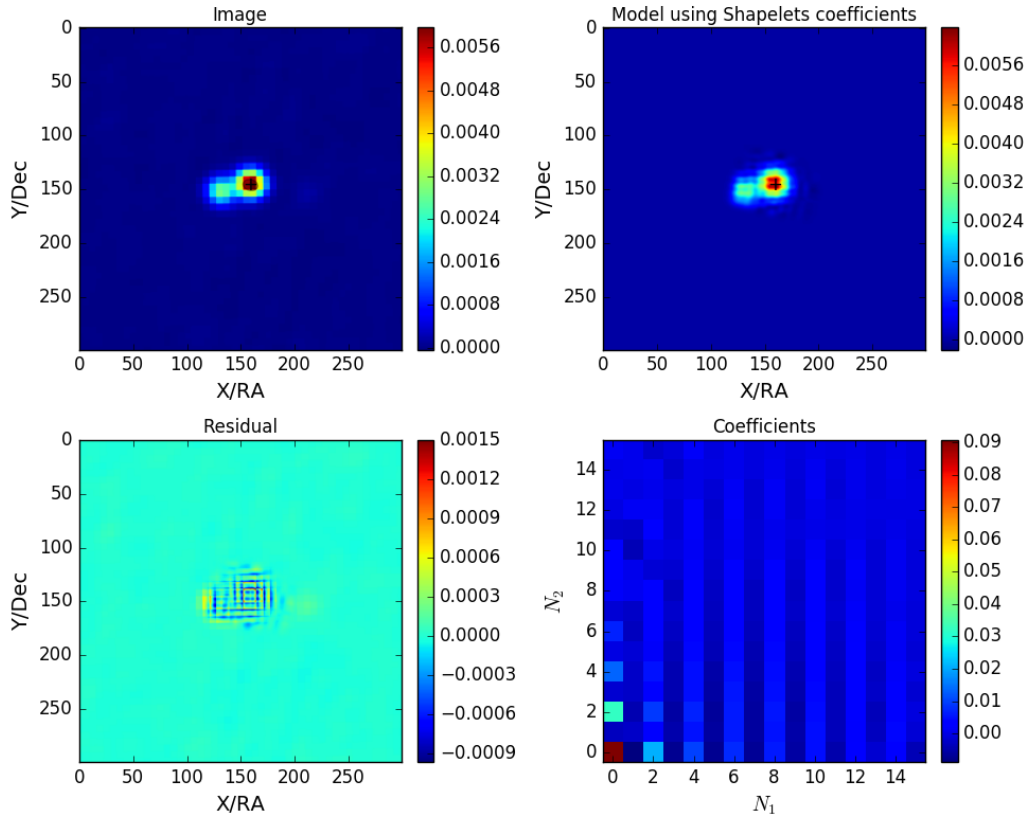


Figure 5.2: MRC0007-287 image (FRI source) decomposed into shapelet coefficients and a model is constructed. The residual is the subtraction of the image and the model.

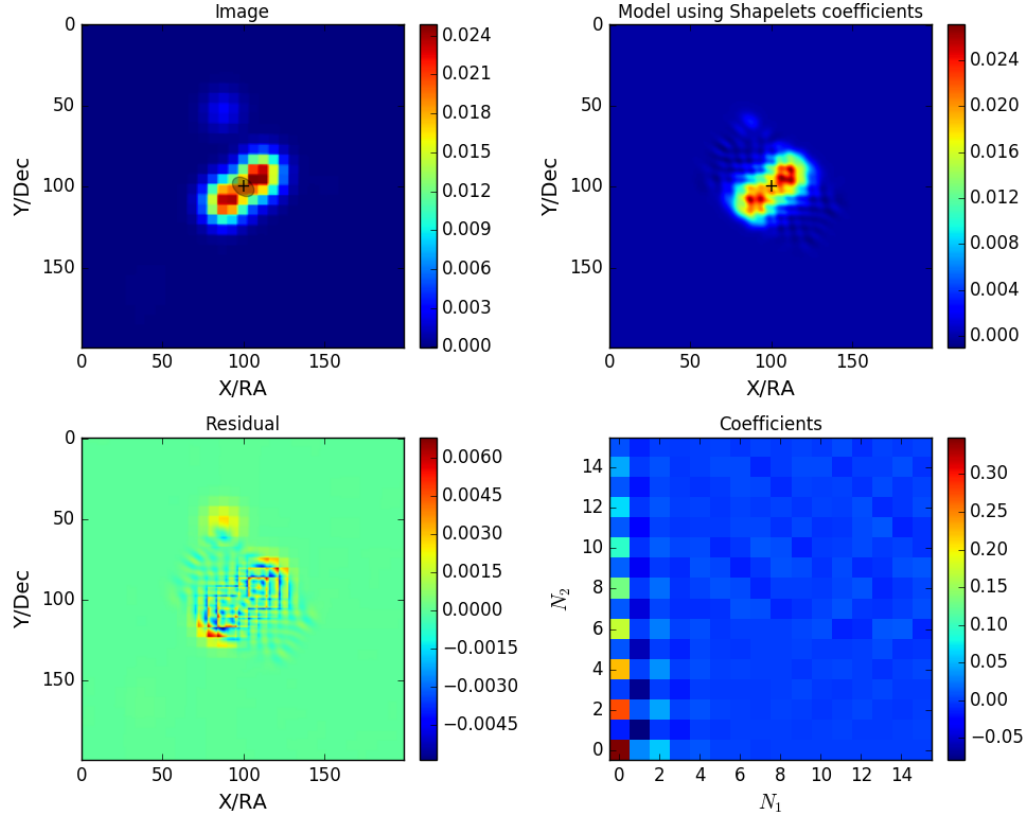


Figure 5.3: MRC0020-253 image (FRII source) decomposed into shapelet coefficients and a model is constructed. The residual is the subtraction of the image and the model.

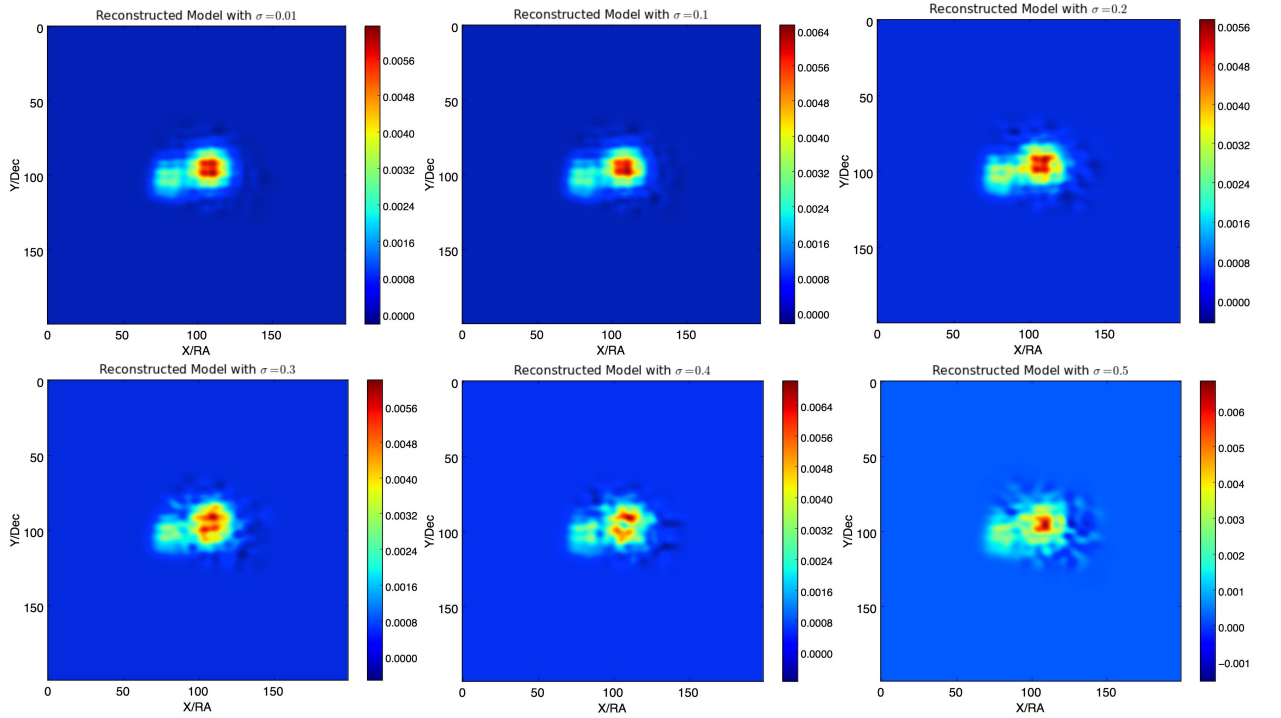


Figure 5.4: Reconstructed models of MRC0007-287 using new sampling coefficients with different σ s.

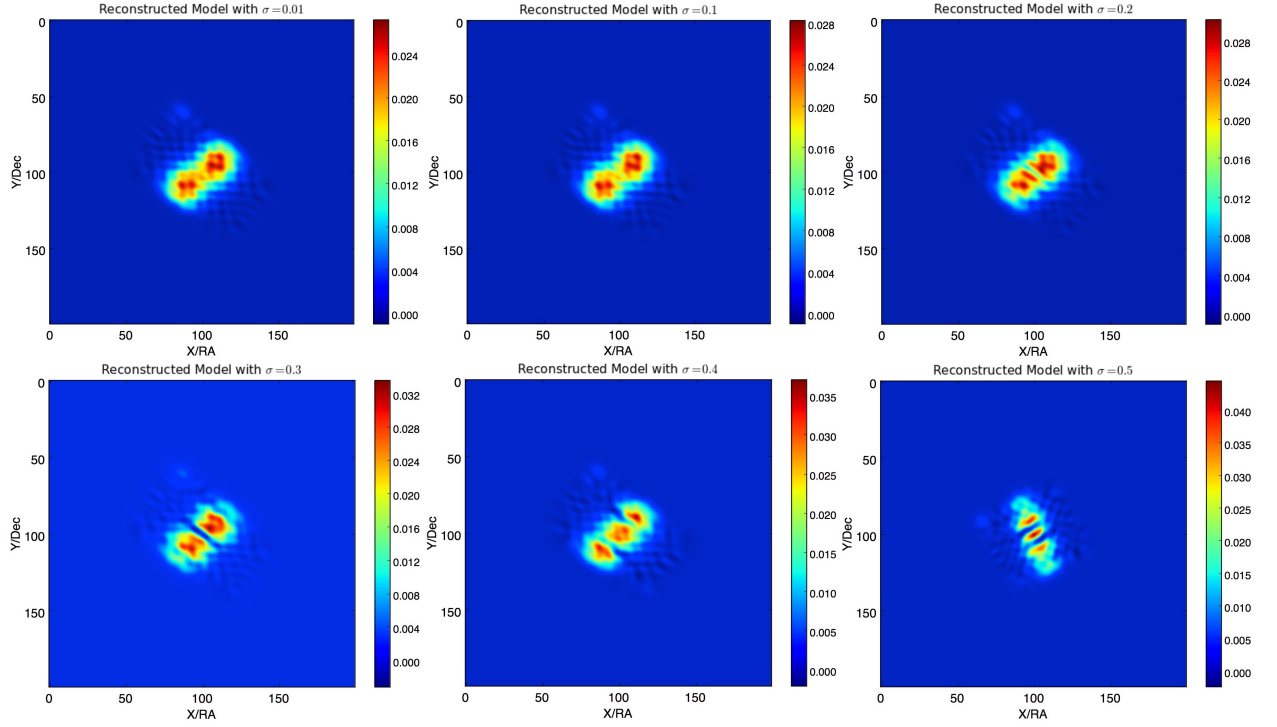


Figure 5.5: Reconstructed model of MRC0020-253 using new sampling coefficients with different σ s. In this case, it is observed that as σ is increased, a good model of the image is not retrieved.

5.4.4 Summary

We have been successful in generating examples of images by sampling through each of the shapelet coefficients of the original image. For each image, the limit in the variation of σ s is different, hence care has to be taken when choosing an appropriate σ for reconstructing images. Choosing a σ outside these limits will result in distorted images and cause the convolutional neural network to learn incorrect features that can bias results. This technique of data augmentation is successful but since we have to look for appropriate σ for each image which is time consuming for large datasets, we will therefore implement a second approach.

5.5 Method 2 : Image pre-processing and Augmentation

Before feeding images into any machine learning algorithm, images need to be pre-processed such that the homogeneity of the image space is maintained. This is a fundamental step for convolutional neural networks as it mimics human visual receptors that capture specific features. For the pre-processing stage, the same method is adopted as in [Aniyan & Thorat \(2017\)](#) that deals with radio images. The background noise and the extent of the flux are estimated using the sigma-clipped statistics⁹ which is an Astropy functionality. Here, a 3σ level is used

⁹http://docs.astropy.org/en/stable/api/astropy.stats.sigma_clip.html#

where pixels above the 3σ from the median are assigned a value of zero. This ensures that unwanted artefacts and background noise are removed. Afterwards, the original images which are of size 300×300 pixels, are cropped from the centre of the images. The process of applying sigma clipping and cut out 150×150 patch is shown in Figure 5.6.

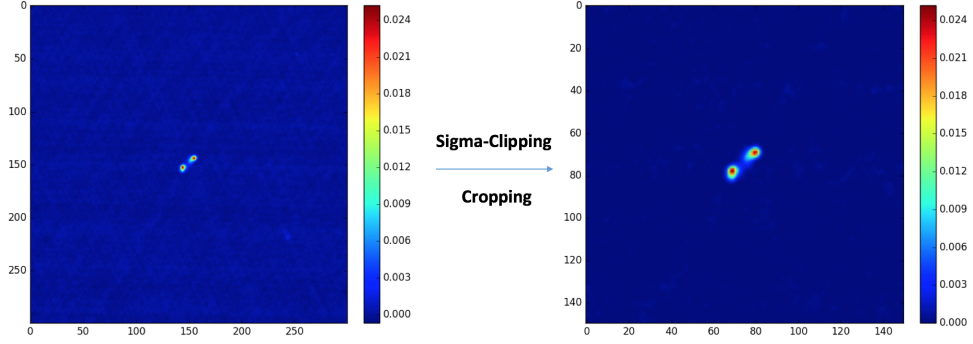


Figure 5.6: Image pre-processing stages where sigma clipping statistics with 3σ to suppress artefacts and noise and a cropping from centre to size 150×150 pixels are applied to the original image with size 300×300 pixels.

Since our dataset is mainly composed of a small sample images of both FRI and FRII, an oversampling technique that preserves the labels that includes rotation, horizontal and vertical flipping is applied to the original existing images. There are many deep learning platforms that construct augmented data. The Keras (Chollet 2016) library was used for performing data augmentation by applying random rotations, scaling and translation transformations during training.

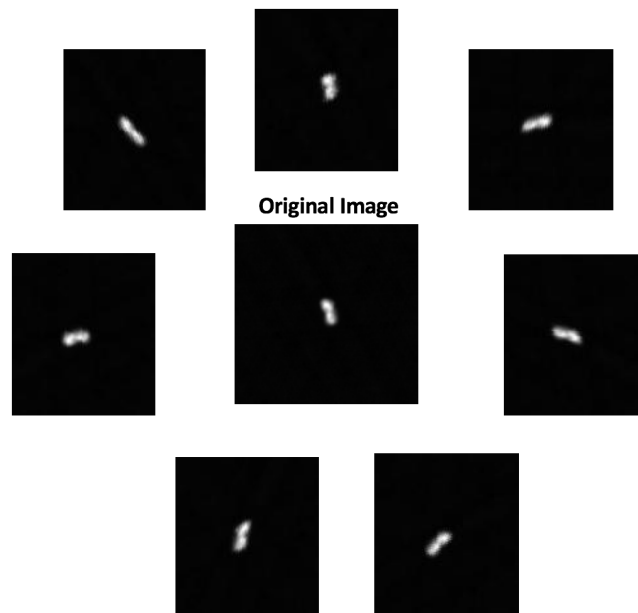


Figure 5.7: The middle panel is the original image of FRI and the images generated are shown all around by applying random rotations, scaling and translation transformations.

Figure 5.7 illustrates an example of image augmentation for an FRI image. A rotational range of 0° to 360° , a vertical and horizontal flipping are applied on the FRI and FRII images resulting in a sample of new images of FRI and FRII where each specific image is taken as a unique training sample. Hence, the problem of imbalanced data samples and overfitting are addressed. Another method for data augmentation is the Deep Convolutional Generative Adversarial Network (DCGAN) which is discussed in next section.

5.6 Method 3: Generating Realistic-Looking Radio Images with Adversarial Neural Networks

Another technique to generate new random sampling in statistics instead of computing the PDF is using a generative model. An introduction about how to train generative models also known as adversarial training is elaborated in detail in this chapter. Adversarial training also known as Generative Adversarial Network (GAN) was proposed originally by Goodfellow et al. (2014). GAN trained two neural networks simultaneously where the first one is called the Discriminator and the second network is known as the Generator.

A GAN consists of a discriminator D and a generator G . The purpose of the discriminator is to discriminate between real and fake images (i.e. images drawn from the true data distribution, like a training dataset, or some other data distribution). The task of the generator is to learn how to produce realistic looking images in such a way that these images will be able to fool the discriminator into thinking that these are real images drawn from the training dataset. The generator produces these images by sampling from a latent ‘prior’ distribution, like a uniform or gaussian distribution. These samples are represented by z , and serve as input to the generator.

When training a GAN, the discriminator is shown a mini-batch of true image samples, x , that are drawn from the training data, as well as a mini-batch of fake/synthetic images produced by the generator, represented by $G(z)$. The goal for the discriminator is to output a single probability p of the images that it has seen as being real. This amounts to the discriminator performing binary classification, where the probability of an image being in the ‘real’ class is p , and the probability of the image being in the ‘fake’ class being $(1 - p)$. The discriminator is then trained to produce probabilities close to 1 for $D(x)$ - i.e. when it is shown actual images from the training dataset. When it is shown fake images produced by the generator, $G(z)$, it is trained to output low probabilities - i.e. $D(G(z))$ should be close to 0.

When training the discriminator, only the discriminator’s weights are updated. For training

the generator, a mini-batch of images from the generator is produced and presented to the discriminator. The generator, however, wants the output of $D(G(z))$ to be close to 1, which translates into fooling the discriminator in thinking that the images it has received from the generator are real. When training the generator in this way, only the generator's weights are updated. As training progresses, the generator should learn how to produce realistic images that resemble images from the training data. In other words, the generator comes close to recovering the true data distribution.

Another method to train generative models with deep learning is Variational Autoencoders. In this project, we focused on a class of Convolutional Neural Networks (CNNs) called Deep Convolutional Generative Adversarial Networks (DCGANs) which is a strong candidate for unsupervised learning.

5.6.1 Generative Adversarial Networks (GANs) building blocks

The building blocks of GAN is represented by a simple distribution defined as p_z which is a uniform distribution varied inclusively between -1 and 1. Then, $z \sim p_z$ is represented by sampling a number from this distribution. Now, we defined $G(z)$ that will take a vector as input and will output an image. This is done using the ideas presented by [Radford et al. \(2015\)](#). In this paper, deep convolutional GANs also known as DCGANs, implemented fractionally-strided convolutions to upsample images which is different from a normal convolution. The latter slides a kernel over an input space (blue) to produce the output space (green) as illustrated in Figure 5.8. In normal convolution, we noticed that the output is smaller than the input.

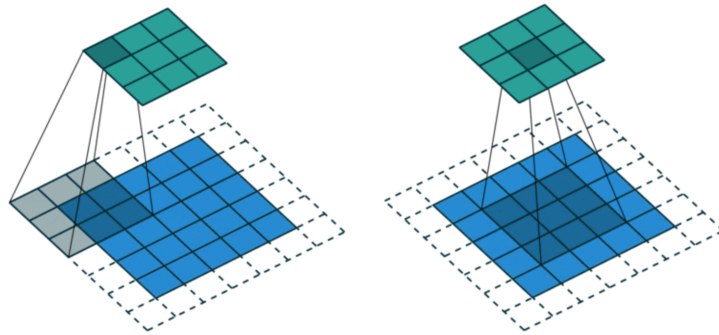


Figure 5.8: Illustration of normal convolution where a kernel slides over an input space (blue) to produce the output space (green). Figure courtesy of [Vdumoulin \(2016\)](#).

Now, fractionally-strided convolution is illustrated in Figure 5.9. Consider an input with 3×3 pixels. The goal is to upsample such that the output is increased. Fractionally-strided convolution is interpreted as the pixels are expanded in such a way that there are zeros in-between

the pixels as shown in Figure 5.9. Afterwards, convolution is carried out on this expanded space that produces a larger output space, here for instance a 5×5 output space is generated.

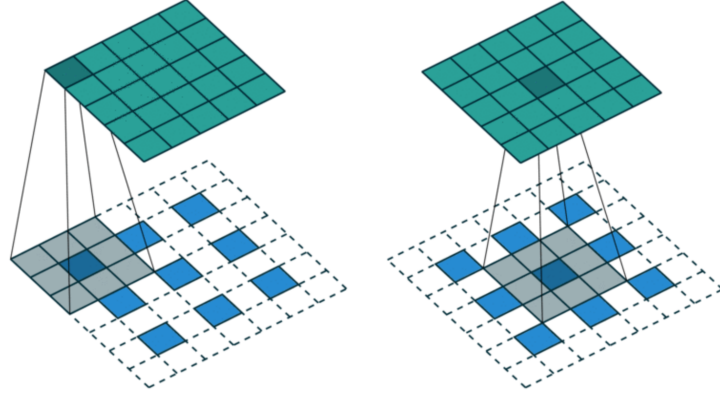


Figure 5.9: A demonstration of fractionally-strided convolution where 3×3 pixels (blue) are expanded with zeros in between the pixels and a normal convolution is carried out to produce an output space of 5×5 (green). Figure courtesy of [Vdumoulin \(2016\)](#).

Since the radio images (FRI and FRII) that we used for training are 150×150 pixels and as building blocks we have the fractionally-strided convolutions, $G(z)$ is therefore represented by the input vector $z \sim p_z$ and produced a $150 \times 150 \times 1$ image. The architecture of the generator is adapted from the work of [Radford et al. \(2015\)](#) and is shown in Figure 5.10 where the first layer of GAN also known as fully connected, takes as input a uniform noise distribution. However, the output is reshaped into a 4-dimensional tensor and it is used at the beginning of the convolution stack. The last convolution layer for the discriminator is flattened and afterwards fed into a single sigmoid output ([Radford et al. 2015](#)).

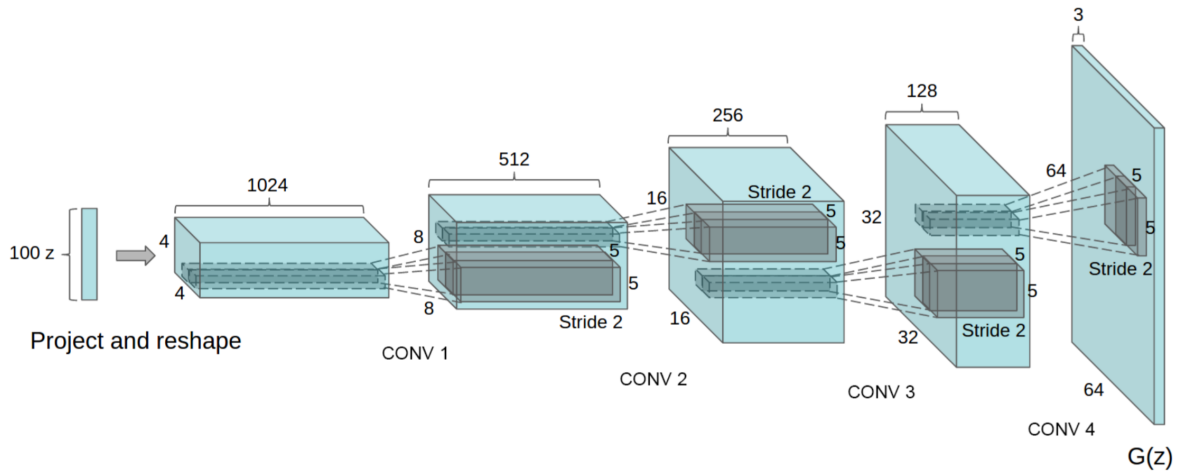


Figure 5.10: The architecture of the DCGAN generator used for the radio image datasets. A uniform distribution z with dimensions 100 is projected to a small spatial extent convolutional representation with many feature maps where a series of four fractionally-strided convolutions then convert this high level representation into a 150×150 pixel image. Figure adapted from [Radford et al. \(2015\)](#)

5.6.2 Approach and Model architectures

Scaling up GANs using CNN has been proved unsuccessful to model images. [Denton et al. \(2015\)](#) had proposed an alternative approach known as LAPGAN to upscale low resolution images and successfully modeled images more reliably. It is known that GANs are unstable to train and many difficulties have been encountered by [Radford et al. \(2015\)](#) to have a stable training process. This work is mostly based on this paper ([Radford et al. 2015](#)) where they adopted three major changes to the CNN architectures for a stable Deep Convolutional GAN. Firstly, all pooling layers acting as a dimensionality reduction technique (see Section 5.8.1.3) in the discriminator and the generator (for example maxpooling layers) have been replaced with strided and fractional-strided convolutions respectively. Secondly, fully connected layers are eliminated for deeper architectures. Thirdly, [Ioffe & Szegedy \(2015b\)](#) proposed a method known as Batch Normalization. This actually stabilized the learning by applying a normalisation on the input to each unit resulting with a mean of zero and variance of one. Therefore, batchnorm is applied to both the generator and discriminator excluding the output layer of the generator and the input layer of the discriminator as it was observed that this had caused some model instability and sample oscillation. Also, in the input layer of the generator, the ReLU activation ([Nair & Hinton 2010b](#)) is applied and the Tanh function is used in the output layer of the generator. For all layers in the discriminator, the leaky rectified activation ([Maas et al. 2013, Xu et al. 2015](#)) is utilized.

5.6.3 Training DCGANs

$G(z)$ is defined and to train it, some parameters need to be found. The unknown probability distribution of the data is denoted as p_{data} . Also, $G(z)$ where $z \sim p_z$ is interpreted as drawing samples from a probability distribution which is also known as generative probability distribution, p_g .

Table 5.2: Some notations of the probability distribution.

| Notation | Meaning |
|------------|--|
| p_z | The sample distribution z |
| p_{data} | The unknown distribution over the images in the dataset and new images are sampled from this distribution. |
| p_g | The generative distribution that the generator G samples from and we actually want to achieve $p_g = p_{data}$ |

In the DCGANs, the discriminator $D(x)$ is a traditional convolutional network illustrated in

Figure 5.11. The discriminator network $D(x)$ used some images x as input and the probability that the image x is sampled from the p_{data} is returned. From the discriminator, a value of 0 is returned when a fake image is generated for example an image produced from p_g and a value near 1 is obtained when the image is from p_{data} .

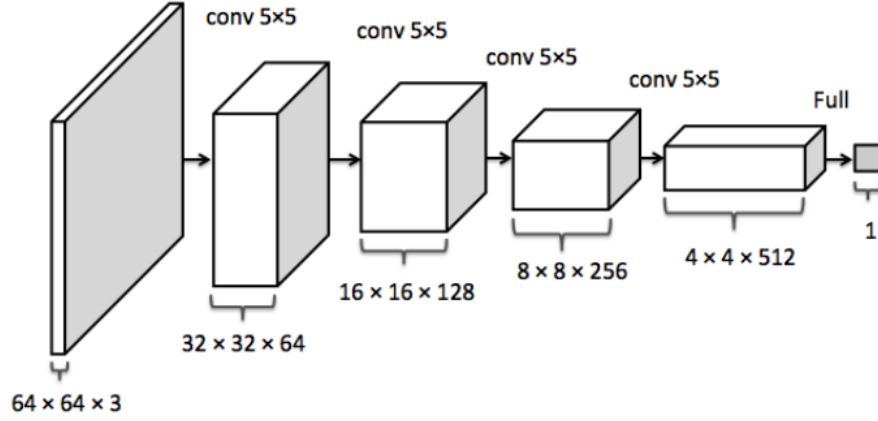


Figure 5.11: The architecture of the discriminator. From left to right, the data is given as an input layer followed with some layers of convolution and end up with a fully connected layer. In our case, the input layer has shape $150 \times 150 \times 1$. Figure from [Yeh et al. \(2016\)](#)

The main goal of training the discriminator $D(x)$ is to maximize $D(x)$ for every image from the true data distribution $x \sim p_{data}$ and minimize it if it is not from the true distribution. The goal of the generator is to generate images that will fool the discriminator. The input of the discriminator is simply an image which is generated by the generator. As a result, $D(G(z))$ is maximized by the generator where D varied between 0 and 1 being a probability estimate. In the paper by [Radford et al. \(2015\)](#), the adversarial network is trained by using the following two-player minimax¹⁰ game with value function $V(G, D)$ given in Equation 5.5 where \mathbb{E} is the expected value.

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))] \quad (5.5)$$

The gradients of Equation 5.5 are taken to train D and G with respect to their parameters. With mini-batches of size m , the expectations are approximated and with k gradients steps, the inner maximization is approximated where it is noted that with $k = 1$, the training performed well.

The parameters for the discriminator and the generator are denoted by θ_d and θ_g respectively. The gradients of the loss with respect to θ_d and θ_g are computed using backpropagation.

¹⁰MinMax stands for maximizing D and minimizing G

The training algorithm is given in Algorithm 2 (Goodfellow et al. 2014). When the training process is completed, that is, $p_g = p_{data}$, it is possible for $G(z)$ to generate new sample from p_{data} .

Algorithm 2 The Training Algorithm

for number of training iterations **do**

for k steps **do**

- Sample mini-batch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample mini-batch of m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{data}(x)$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[\log D(x^{(i)}) + \log (1 - D(G(z^{(i)}))) \right]$$

end for

- Sample mini-batch of m noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(z^{(i)})))$$

end for

5.6.4 Details of Dataset for DCGAN Training

The DCGAN is trained on two separate datasets: FRI and FRII radio images as summarized in Table 5.1 where the sigma-clipping statistics with 3σ level and a cropping are implemented as pre-processing. One requirement of neural networks is to have a large number of training examples. Therefore, the FRI and FRII images are augmented using the techniques of rotating and flipping as described in Section 5.5 where 20 000 FRI and 20 000 FRII images are created for training the DCGAN. The images of 150×150 pixels are used as input for the training. However, they are scaled with a tanh activation function with a range of $[-1, 1]$. With mini-batch stochastic gradient descent (SGD), all models were trained with a mini-batch size of 100. In addition, same parameters and weights are used as in Radford et al. (2015). A zero-centered normal distribution with standard deviation 0.02 was utilized as initialization for all weights. In all models, the slope of the leak in the LeakyReLU (see Section 5.8.1.4) was established to 0.2. Moreover, the Adam optimizer (Kingma & Ba 2014) with tuned hyper-parameters and a learning rate of 0.0002 are used. Adam optimizer derived from adaptive moment estimation is a method for efficient stochastic optimization that only requires first-order gradients. The

method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients (Kingma & Ba 2014). To stabilize the training, the momentum term β_1 is reduced to 0.5. Momentum is a method which helps accelerate gradients vectors in the right directions, thus leading to faster converging.

5.6.5 Generating FRI and FRII radio images using DCGAN

In the Radford et al. (2015) paper, the DCGAN is trained on a dataset of bedroom images. In this project, the network is trained using FRI and FRII datasets and then realistic looking FRI and FRII Radio images are generated. The code has been adapted from Taehoon Kim's¹¹ repository which is implemented in a Python class. The code has been modified for radio images.

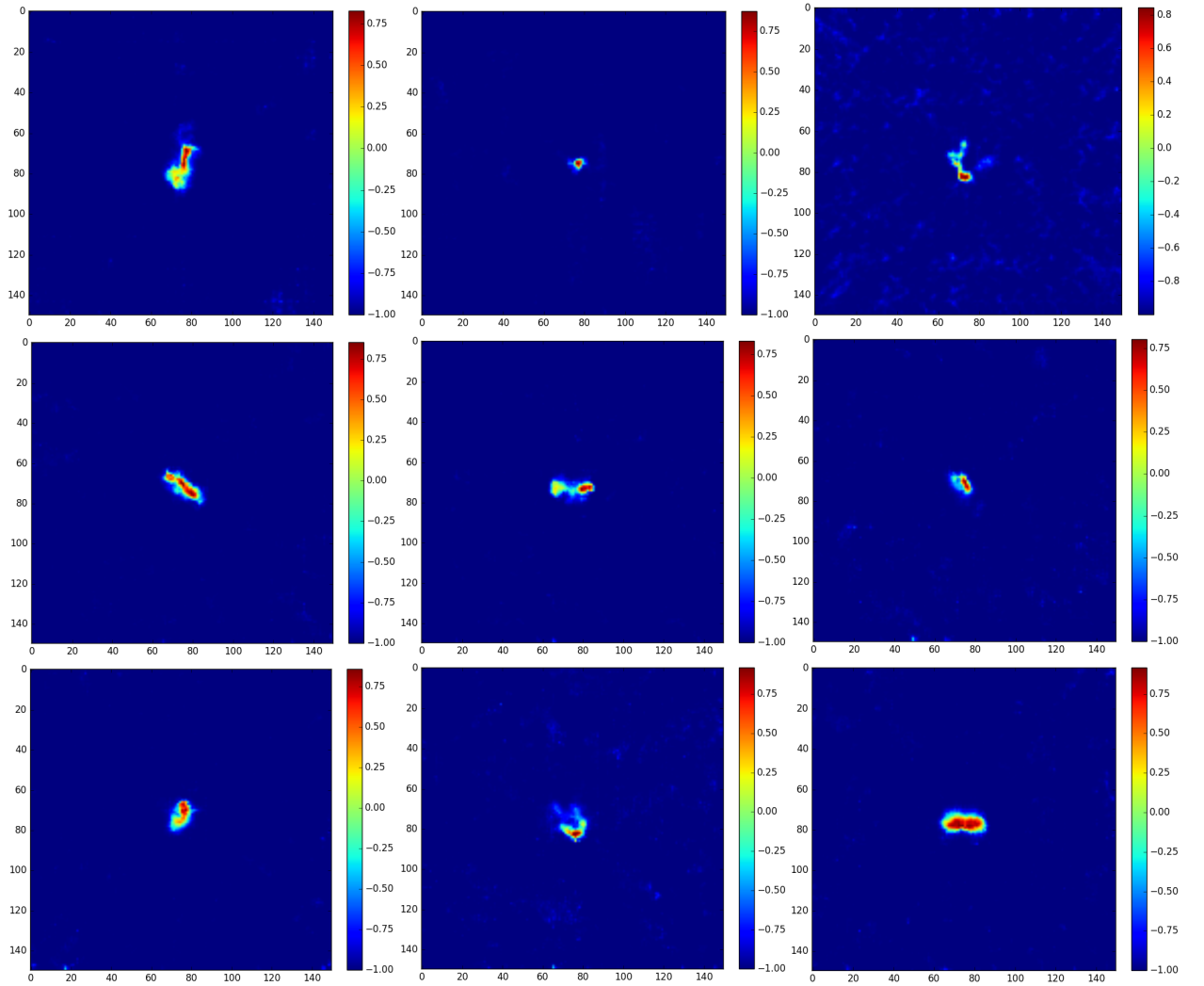


Figure 5.12: FRI simulated radio images from DCGAN trained with 30 epochs and batchsize 100 using (150×150) pixel images of 20 000 FRI images.

For 30 epochs of training, samples of FRI and FRII images after convergence are shown in Figure 5.12 and Figure 5.13. Henceforth, these simulated images are called DCGAN FRI and

¹¹<https://github.com/carpedm20/DCGAN-tensorflow>

DCGAN FRII.

It is observed from Figure 5.12 and Figure 5.13 that there are repeated noise features across various samples and these simulated images are good examples of FRI and FRII sources. We note that DCGAN FRIs have one hotspot near the core while DCGAN FRIIs have two distinct hotspots at the end, thus showing a good representation of the radio sources that can be used for the classification.

Figure 5.13 shows some samples generated after training FRII radio images with 30 epochs. It is seen that similar visual appearance samples are generated when compared to the original FRII radio images.

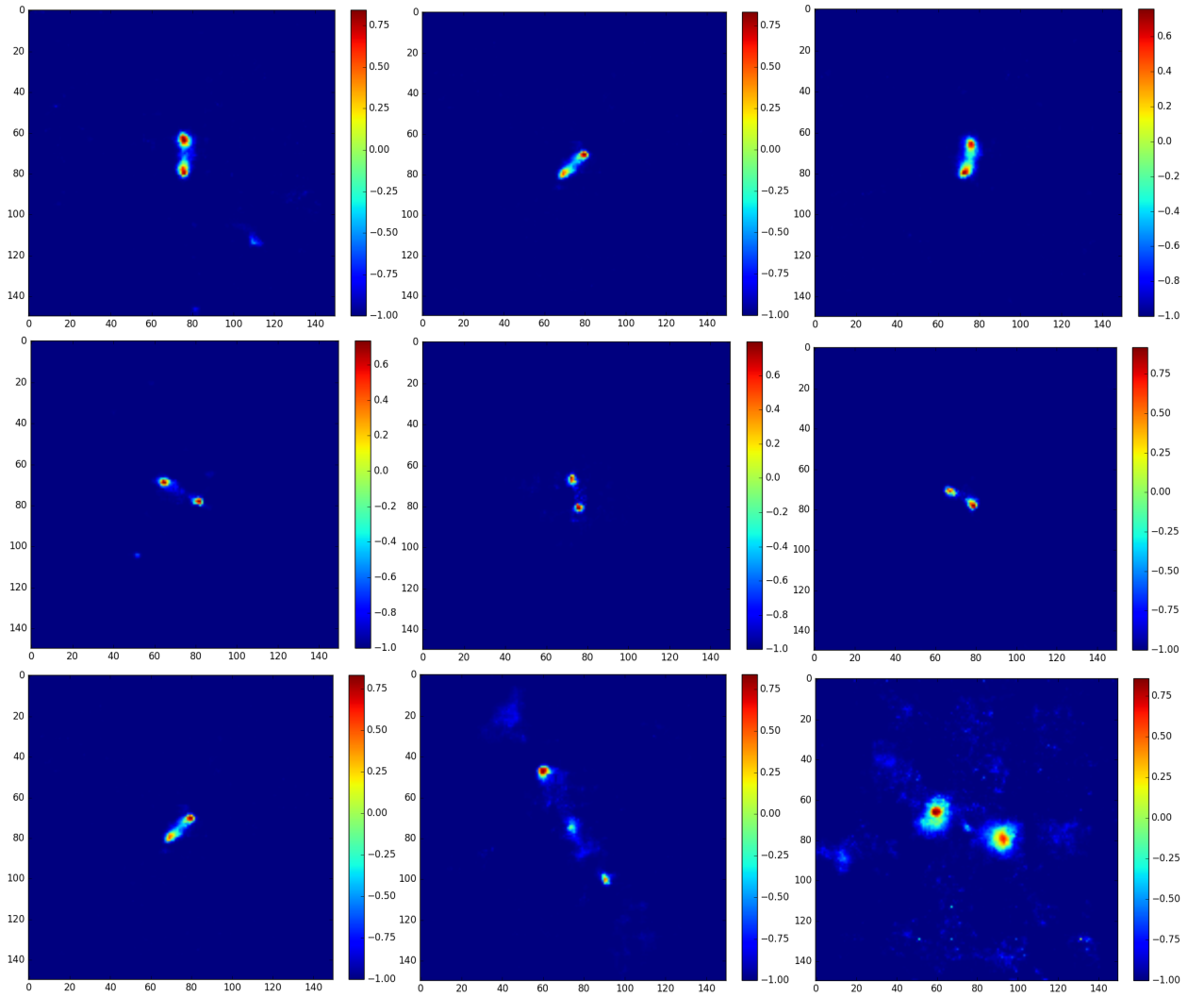


Figure 5.13: FRII simulated radio images from DCGAN with 30 epochs and batchsize 100 using (150×150) pixel images of FRII images. It is noted that model learned to memorize the training samples, hence good examples of FRII images are generated.

This study shows the first application of GANs to radio astronomy. Training on these image datasets (FRI and FRII datasets), convincing evidence is shown that adversarial networks learned a hierarchy of representations of images in both the generator and discriminator for

supervised learning.

5.7 Introduction to Convolutional Neural Network

In the last few years, breakthrough results were led by deep neural networks in the field of pattern recognition tasks such as voice/speech recognition (Hinton et al. 2012) and computer vision. A special form of neural network has led to these results which are called the Convolutional Neural Network (CNN) or ConvNets. These networks have been intensively used for image classification (Lawrence et al. 1997) and they can be used to classify images better than humans into specific categories for instance digit recognition was a successful application of CNN by LeCun & Bengio (1995). In some basic way, ordinary Neural Networks are introduced as shown in Figure 5.14.

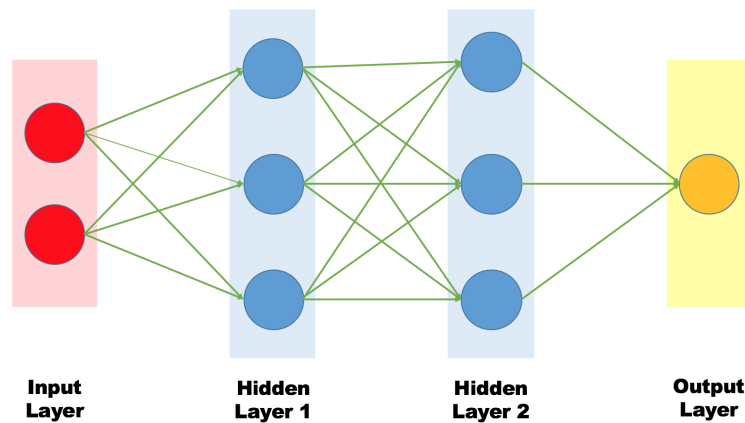


Figure 5.14: An illustration of a regular 3-layers Neural Network where the circles represent the neurons in each layer.

A neural network receives a single vector as input and the data is altered through a series of hidden layers. From Figure 5.14, it is observed that each hidden layer is made up of a series of neurons. Each neuron in the hidden layers are fully connected to the previous layer. In addition, it is seen for each layer, neurons are completely independent having no connection among each other. The last fully connected layer is known as the output layer and in the classification scheme, it is used to obtain the output of the network which can be used to determine the predicted class.

As we have seen, a neural network is made up of neurons that learned and updated the weights and biases of the models. Some inputs are given to each neuron and a dot product is performed. Then, optionally an activation function with a non-linearity is followed. At the end of the network, there is a loss function (e.g. SVM/Softmax) and this layer is called the last fully-connected layer. In a mathematical way, the output z of a single neuron can be represented

as

$$z = \sum_{i=1}^d w_i x_i + b_0 \quad (5.6)$$

where the different inputs to the neuron is given as x_i , w_i are the weights to the inputs and b_0 is the bias. $w_i x_i$ is simply a dot product. Afterwards, the output is given to an activation function as follows

$$\hat{z} = f(z) \quad (5.7)$$

where the activation function is represented as f . Examples of activation functions are the Tanh function and the sigmoid function (see Section 5.8.1.4). Therefore, Equation 5.6 is rewritten as

$$\text{Network}_j = \sum_i^{N_H} z_i w_{ji} + b_{j0} \quad (5.8)$$

where j represents each unit in the output layer and the number of hidden layers in the network is given as N_H .

5.8 Convolutional Neural Network architecture

The input layer of a CNN takes an input with a known width, height and depth (e.g. $30 \times 30 \times 3$). Unlike a regular neural network, in particular, CNN have their neurons grouped in three dimensions (width, height and depth) and are connected to previous small part of the layer as seen in Figure 5.15. At the end of the CNN architecture is the output layer that reduces the input image into a single vector of class scores.

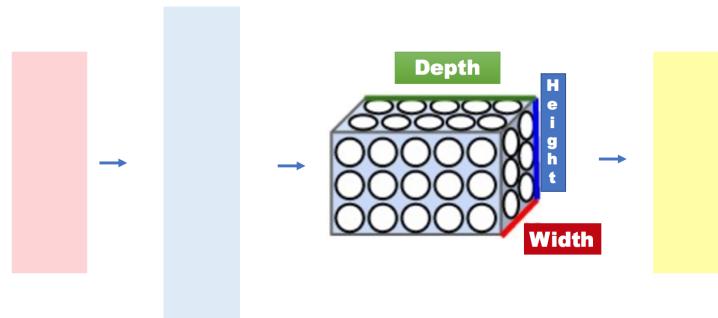


Figure 5.15: In one of the layers, the CNN displays the neurons in three dimensions (width, height and depth). In the CNN, each layer transforms the input (3D volume) to a 3D output volume of neuron activation. The input layer is shown in Red for e.g an image with its width and height and the 3 channels (Red, Blue and Green) will be the depth.

For CNN, the dot product in Equation 5.8 is replaced by a convolutional operator given as

$$\text{Network}_j = \sum_i^{N_H} z_i * w_{ji} + b_{j0} \quad (5.9)$$

where $*$ represents the two-dimensional convolution operation and w_i represents the filter or a kernel. This filter allows CNN to learn on raw data directly rather than on features designed by astronomers. Traditionally, features have to be defined which are inputs into a NN, whereas with CNNs, features do not need to be designed. The filters learn the features automatically thus providing one of the main advantages of CNNs. Different features are learnt in each layer of convolution. For instance, simple features like edges, outlines and corners are learnt in the first few layers. Afterwards, a combination of those elementary features in each successive layers led to more complex features that describe the input data, hence provide a hierarchy of features.

5.8.1 Classification using CNN

One of the main objectives of this chapter is to design a CNN model to classify FRI and FRII sources. The task of CNN is that each time it is given a picture, it has to decide whether it is an FRI or an FRII source as shown in Figure 5.16.

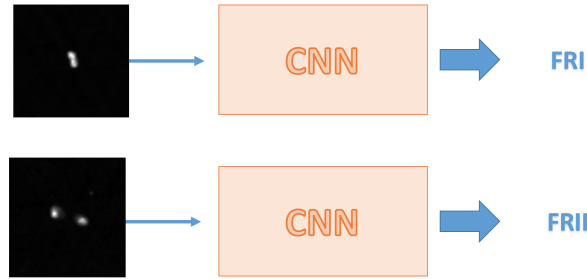


Figure 5.16: Classification of FRI and FRII using CNN.

An image to a computer is viewed as an array of pixels where at each position a number is attributed. In this case, a pixel value of 1 is given to the white pixels and -1 to the black pixels. If any values of the pixels do not match when two images are compared, then to a computer, these two images are different or are not of the same class as shown in Figure 5.17. Ideally, we want to have an algorithm that will be able to classify an FRI and FRII source even if the sources are shifted, flipped, rotated or deformed.

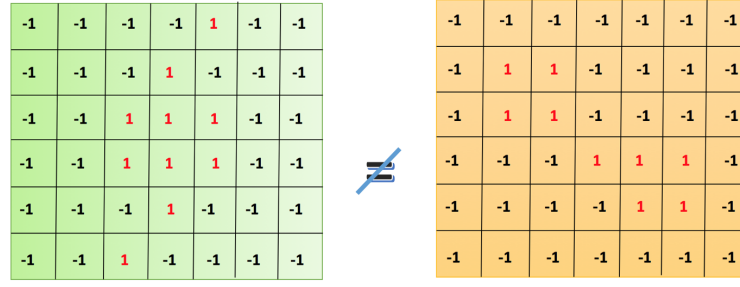


Figure 5.17: Illustration of pixel values for two different images (FRI and FR II).

5.8.1.1 Features

CNN takes an input image, applies the filters which generate features. CNN locate features that roughly matched the same locations in two images where each feature is viewed as a small 2D array of values (mini image). In the case of FRI, features having a compact shape capture all the fundamental characteristics of most FRI sources while for FR II there are two distinct hotspot features that are attributed to capture FR II sources in an image as shown in Figure 5.18.

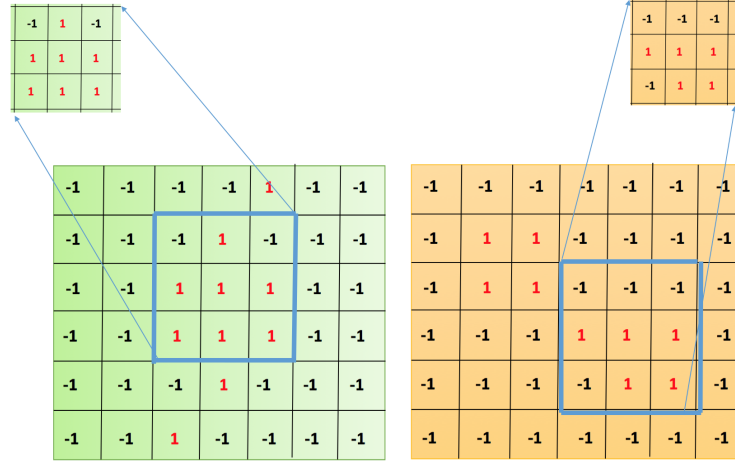


Figure 5.18: The features allocated for FRI and FR II source.

One important aspect of CNN is weight sharing where translational independent features are extracted from the input data. In our case, we consider an FRI source as an example. The input image is still an FRI even though the FRI source may be located at the centre or at the different locations in the image. The CNN learns specific features that make a galaxy an FRI or an FR II even though the sources are flipped, rotated, or at different location in the images. Therefore, through weight sharing the CNN learns features that are flipped, translated and rotated. Hence, the weights for a specific class irrespective of rotation and flipping are the same and are shared among the different samples in the same class. Hence, CNN is an ideal

algorithm to be applied for this problem. We use three main types of layers to build ConvNet architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer. We will stack these layers to form a full ConvNet architecture and these layers are explained in the next sections.

5.8.1.2 Convolutional Layer

When a new image is presented to the CNN, the latter is not aware where the features will correspond to. Therefore, the CNN strides the features across the whole image in every possible position and this process is known as filtering. It is simply the convolution of the features to the entire image. Convolution is the multiplication of each pixel in the feature to the value of the corresponding pixel in the image. Afterwards, these values are added together and are then divided by the total number of pixels in the feature. A value of 1 is obtained when both pixels are white ($1 \times 1 = 1$) and are black ($-1 \times -1 = 1$). As a result, any mismatch will result in a value of -1 . The process of convolution is illustrated in Figure 5.19. This process is repeated by striding the feature with every possible patch of the image where a new 2D array is formed to store each answer from each convolution, also called the 'Convolved Feature', 'Activation Map' or the 'Feature Map'. A value of 1 shows strong matches and a value close to 0 shows no match of any sort.

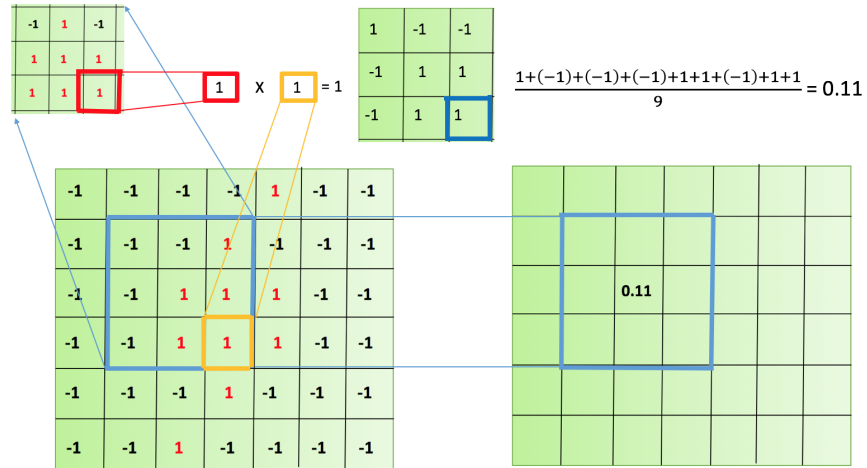


Figure 5.19: A demonstration of convolution. The feature (3×3) strides over the image (6×7) where a convolution is performed to give a value near 1 for strong similarities and 0 for any sort of mismatch.

5.8.1.3 Max Pooling Layer

Pooling in the CNN acts as a dimensionality reduction technique. It reduces large images to smaller images, while still preserving the most fundamental features in them, i.e preserving the best fits of each feature within the window. Max pooling utilizes a small window and strides it

over the image by taking the maximum value at each step along the image as shown in Figure 5.20.

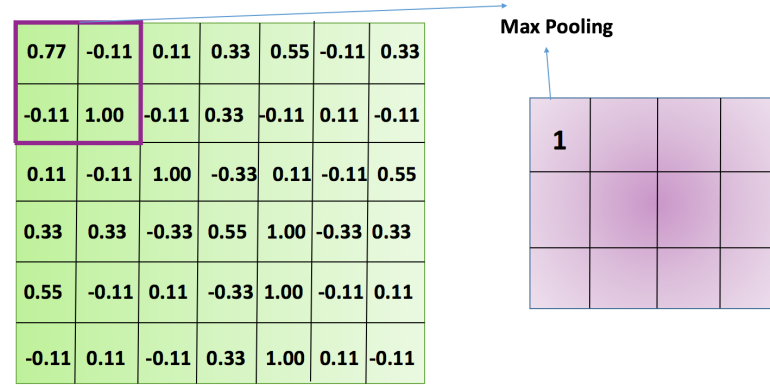


Figure 5.20: Application of Max Pooling on an image. The image (6×7) pixels is reduced to (3×4) pixels.

5.8.1.4 Rectified Linear Units and the sigmoid function

In the recent literature, two activation functions are commonly used by CNNs (Nair & Hinton 2010a, Krizhevsky et al. 2012). They are the sigmoid function and the rectified linear unit (ReLU) as shown in Figure 5.21.

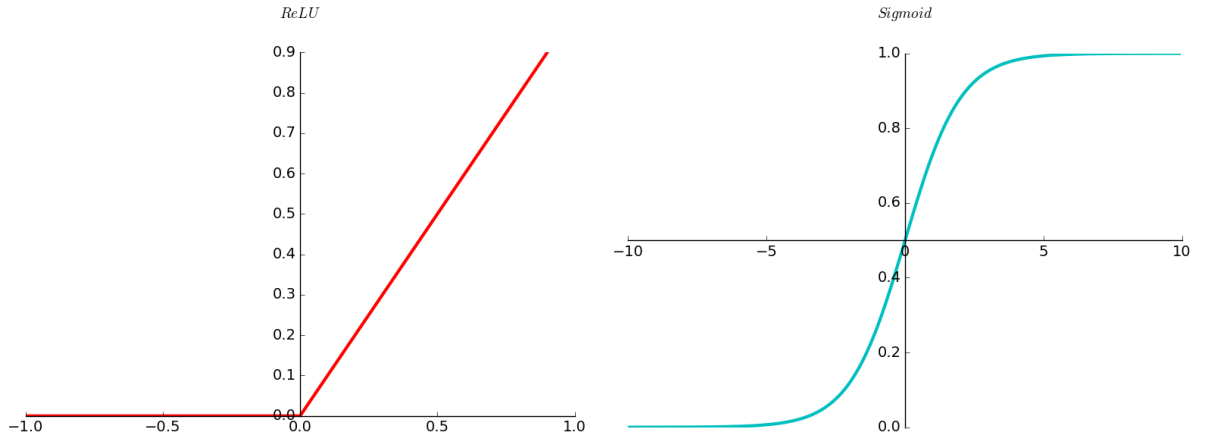


Figure 5.21: Two nonlinear activation functions adopted by CNNs: the ReLU (Left) and the sigmoid function (Right).

All of them play a clipping-like operation. This is a layer of neurons that applies the non-saturating activation function $f(x) = \max(0, x)$. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. Other functions are also used to increase nonlinearity, for example the saturating hyperbolic tangent $f(x) = \tanh(x)$ and the sigmoid function $f(x) = (1 + e^{-x})^{-1}$. The sigmoid clips the input into an interval between 0 and 1. The ReLU clips negative values to zero while keeping positive values unchanged, i.e, it changes a value to 0 whenever it encounters a

negative number. Leaky ReLUs allow a small, non-zero gradient when the unit is not active. Instead of the function being zero when $x < 0$, a leaky ReLU will instead have a small negative slope of 0.01 for example. That is, the function computes $f(x) = \max(x, ax)$ where a is a small constant that is less than 1.

Compared to other functions the usage of ReLU is preferred and beneficial for CNN, because it results in faster training time without making a significant difference to generalisation accuracy and from getting stuck near 0 or tending towards infinity.

5.8.1.5 Fully Connected Layer

The last layer of CNN is the fully-connected (FC) layer. The high-level filtered images act as input for the FC where the latter further translates them into votes. The filtered images being two dimensional arrays are treated as a single list by FC. When a new image is given as input to the CNN, it passes through the first few layers until it arrives at the end of the fully connected layer. In this project, our aim is to classify between the two types of sources FRI and FR II. The filtered images being two dimensional arrays are treated as a single list by FL. Whether the image is an FRI or FR II, every image obtains its own vote.

5.8.2 Network model and training

For this study, different models were written in Python 3.5.3 using different libraries such as Keras (Chollet 2016), which is a Deep Learning library using Theano as backend and TFLearn¹² which is a deep learning library built on top of Tensorflow. The network for this study was implemented in Tensorflow (Abadi et al. 2015) another deep learning package widely used in the field of computer vision. The main objective of this chapter is to develop new techniques to perform classification of FRI and FR II radio galaxies.

The overall architecture of our ConvNet is presented in Table 5.3. The network comprises of eight trainable layers with 5 convolutional layers, 3 layers of maxpooling and the output of the final maxpooling layer is passed to 3 fully connected layers each followed by ReLU activations. In addition, each convolutional layer is followed by an activation layer (ReLU) and the first two convolutional layers are followed by a batch normalization. A demonstration of the ConvNet is shown in Figure 5.22.

There are weak connections among the neurons within the fully connected layers that are insensitive to weight updates and these weak connections can be discarded since they have no influence on the forward pass. This mechanism is known as *dropout* (Hinton et al. 2012) and

¹²<http://tflearn.org/>

it is a technique to avoid overfitting. In the ConvNet, a dropout of 0.5 is applied, i.e, 50% of the weak neuron connections are removed, thus allowing the network to learn more robust features.

Table 5.3: A summary the ConvNet architecture.

| Type | Filters | Filter Size | Activation Function |
|-----------------|---------|----------------|---------------------|
| Convolution | 6 | 11×11 | ReLU |
| Max Pooling | 6 | 3×3 | - |
| Convolution | 19 | 5×5 | ReLU |
| Max Pooling | 19 | 3×3 | - |
| Convolution | 38 | 3×3 | ReLU |
| Convolution | 26 | 3×3 | ReLU |
| Convolution | 26 | 3×3 | ReLU |
| Max Pooling | 26 | 3×3 | - |
| Fully-Connected | 40 | - | ReLU |
| Fully-Connected | 40 | - | ReLU |
| Fully-Connected | 2 | - | Softmax |

During training, the loss is calculated from the cross entropy error which is given as

$$L(w) = -\frac{1}{N} \sum_{n=1}^N [y_n \log \hat{y}_n + (1 - y_n) \log (1 - \hat{y}_n)] \quad (5.10)$$

where the output during the forward pass is given by y_n , w is the weight, N is the number of training samples and \hat{y} is the expected output (Aniyan & Thorat 2017). The last fully connected layer has a depth of two that calculates the cross entropy loss and outputs a softmax probability score during validation for the two categories. Each layer is described in Table 5.3 with their corresponding size and activation function used.

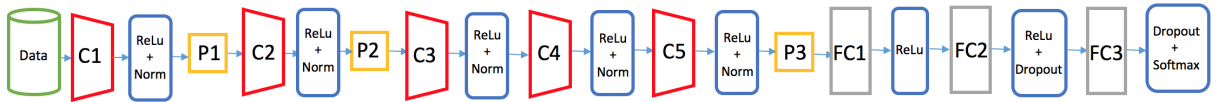


Figure 5.22: ConvNet architecture implemented for this study. The network consists of 8 trainable layers with 5 convolutional layers (C), 3 max pooling layers (P) and 3 fully connected layers (FC).

For training our ConvNet model, the final dataset in Table 5.1 is split into 50% for training, 25% for validation and the last 25% for testing where in general practice a larger portion is taken for training. Therefore, the actual sample number for training is 48 FRI and 185 FRII. The

validation set consists of 24 FRIs and 92 FRIIs. And the samples in the validation and testing set have not been seen by the model during training. The data is augmented using the second method (i.e. application of rotation and flipping) and the third method for oversampling, (i.e, the DCGAN).

Table 5.4: A summary of the dataset size used for training the ConvNet model.

| Samples | Types of Sources | Final Sample Size | Final samples using Flipping & Rotation | Final samples using DCGAN |
|------------------|------------------|-------------------|---|---------------------------|
| | | | CASE 1 | CASE 2 |
| Training (50%) | FRI | 48 | 10000 | 10000 |
| | FRII | 185 | 10000 | 10000 |
| Validation (25%) | FRI | 24 | 3000 | 3000 |
| | FRII | 92 | 3000 | 3000 |
| Testing (25%) | FRI | 24 | - | - |
| | FRII | 92 | - | - |

A summary of the data set used for training our model is given in Table 5.4. Firstly, we trained the model with datasets that have been augmented using rotation and flipping. For Case 1, we used 10000 FRIs and 10000 FRIIs as training sets, being validated with 3000 FRIs and 3000 FRIIs. Afterwards, the model is tested with original images of FRIs (24) and FRIIs (92) in the test samples.

Secondly, we trained the model with DCGAN images. For Case 2, we used 10000 DCGAN FRIs and 10000 DCGAN FRIIs as training sets, then validated with 3000 DCGAN FRIs and 3000 DCGAN FRIIs. The model for Case 2 is tested with original images of FRIs (24) and FRIIs (92) in the test samples as given in Table 5.4.

The training for both cases was done with training steps of 1000 and with a batch size of 50 using ADAM (Kingma & Ba 2015) as an optimizer to minimize the loss with an exponential decay rate of 0.5 and a decay learning rate of $\alpha = 0.0001$. During training, a large number of hyper-parameters has to be learnt by the network, thus a large amount of memory is required which is one drawback of deep neural network. Therefore, to overcome the memory requirements we used four NVIDIA TITAN-X Black GPUs each having 12GB of RAM.

5.9 Results and Analysis

In this work, with the two classes of sources, the ConvNet is trained for a binary FRI-FRII classification. The performance of the model is evaluated using the following metrics: AUC, accuracy, precision, recall and F1-score (see Section 4.5.2). The trained model was used to perform classification on the testing samples (i.e. on original images). The model is trained from scratch separately for the two cases and the results for the classification are given for the two cases in Table 5.5. We observe that the model for Case 2 shows excellent performance with DCGAN images (Case 1) compared to the model trained with rotated and flipped data (Case 2). For Case 1, we obtain an AUC value of 0.80, an average precision of 78% and average recall of 57% with an F1 score of 61%. Having a low recall and F1 score imply that the algorithm for Case 1 was not able to identify the differences between FRI-FRII radio galaxies.

Table 5.5: A summary of the performance results of the ConvNet model for FRI-FRII classification.

| DATA | Precision | Recall | F1-Score | Accuracy | AUC |
|--------------------------------|-----------|--------|----------|----------|------|
| CASE 1 (Flipping & Rotated) | 0.78 | 0.57 | 0.61 | 0.57 | 0.80 |
| CASE 2 (DCGAN) | 0.83 | 0.84 | 0.83 | 0.84 | 0.85 |

For Case 2, the model performs the classification of FRI-FRII with an accuracy of 84%, average precision is 83%, average recall of 84% with an F1 score of 83%. We also present the ROC curve for this network in Figure 5.23 where the true positive rate is plotted against the false positive rate. We obtained an AUC of 0.85 that quantifies the overall performance of the ConvNet. We observe that training the algorithm with DCGAN images shows an excellent improvement in the classification of FRI and FRII radio galaxies. Case 2 shows a very high precision, recall and F1-score, meaning that most classification of FRI and FRII has been correctly identified.

In Figure 5.24, we show some examples of our testing samples where the FRI sources display more than one hotspot. There may be a possibility that the model is confused by the bright hotspots and diffuse emission that leads to some misclassification of the two classes of sources. The overall performance of the model outputs a good precision and recall, hence we can conclude that without much confusion, the model for Case 2 is able to classify FRI-FRII radio galaxies.

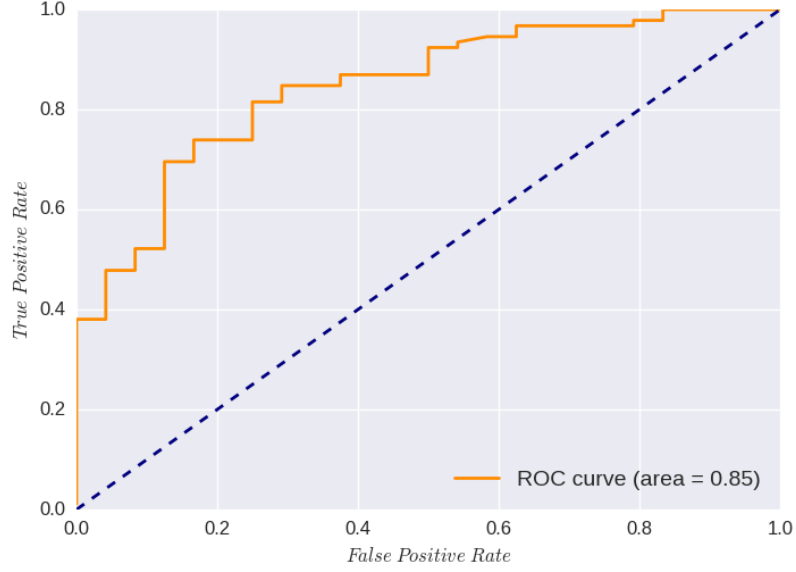


Figure 5.23: The ROC curve and the AUC value of 0.85 is indicated in parentheses for the ConvNet classifier using DCGAN images.

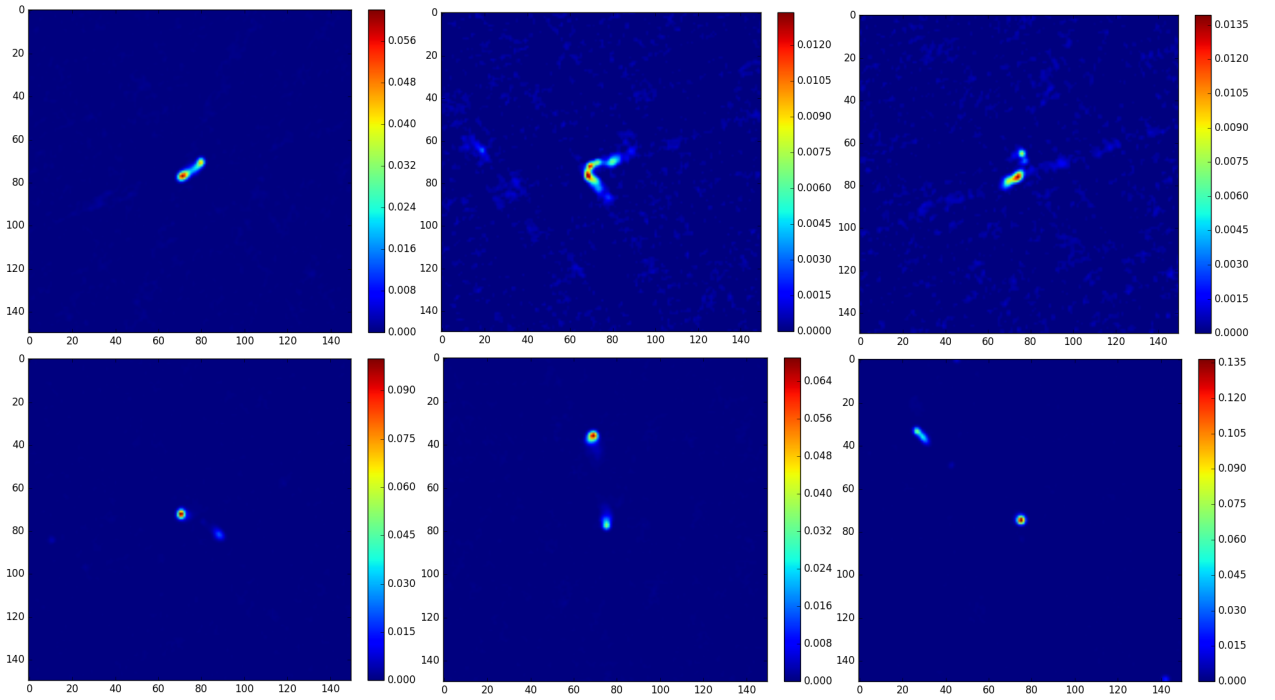


Figure 5.24: Some examples images in the testing samples. The first row are examples of FRI sources and the second row are examples of FRII sources. It is observed that these FRI sources have mostly similar characteristics as FRII sources and vice versa for the FRII sources, thus create a confusion for the model to perform the correct classification.

5.10 Summary

In this Chapter, we have implemented three methods to artificially increase the number of training data, hence overcoming the problem of overfitting. We implemented a first method to

sample through shapelets coefficients with varied σ s. Secondly, we applied a rotational range of 0° to 360° , horizontal and vertical reflection on each training samples. In the third method, we introduced the first application of deep convolutional generative adversarial networks (DCGANs) to generate FRI and FRII radio galaxies.

Afterwards, we trained a CNN model for two cases: i) Using flipped & rotated data , ii) Using DCGAN FRIs and FRIIs as a training and validation set. We observed that training the network with DCGAN images shows an excellent improvement in the classification of FRI-FRII radio galaxies.

[Aniyan & Thorat \(2017\)](#) have implemented a deep convolutional neural network along with a fusion classifier to perform binary classification of FRI-FRII radio galaxies, FRI-bent radio galaxies, and FRII-bent radio galaxies. Their training sample for each category is around 200 sources which have been augmented using rotation and flipping. They obtain an individual precision of 95% for bent radio galaxies, 91% for FRI and 75% for FRII classes. We can not actually compare our results with [Aniyan & Thorat \(2017\)](#) as they have used a different network and they implemented a fusion classifier to obtain individual precision for each class. The sample data they have used are different from ours and they have used a different augmentation strategy. And the total amount of augmentation data they have used are different from ours.

In our case we have performed only a binary classification between FRI and FRII sources using DCGAN as an augmentation technique. We have been successful in classifying FRI-FRII sources with an accuracy of 84% compared to the machine learning algorithms applied in Chapter 4. To conclude, our results reveal that a combination of GAN and CNN add significant value to the classification of FRI and FRII galaxies and the results are comparable to human performance.

Chapter 6

6 Conclusion

We are entering the new paradigm of very high volumes of data with the SKA and its precursors. We are expecting to have petabytes of data with large source density per square degree for surveys. Therefore, this deluge of data is next to impossible to handle through manual techniques. Hence, it is essential to have automated data processing.

The main objective of this thesis is to develop new techniques to perform classification of FRI and FRII radio galaxies and to distinguish between Point and Extended sources. In Chapter 1, a brief introduction about radio galaxies, and different types of astronomical sources was presented. Different techniques for classification of astronomical sources were introduced, focusing on filtering methods. We also provided an overview of some surveys and the data used for this work.

In Chapter 2, the field of machine learning, we provided a theoretical background for the different algorithms we employed, including the k -Nearest Neighbours (k NN), the Random Forest (RF), the Multi-Layer Perceptron (MLP) and the Naive Bayes (NB) as classification algorithms, and the Principal Component Analysis (PCA) as dimensionality reduction techniques were discussed. In this work, we mainly focused on supervised learning where the true label of each training source is known.

The main aim of Chapter 3 was to detect only extended sources in some samples of the SUMSS catalogue. An image analysis approach was performed where the source properties: flux, area, background noise, and the centroid of sources are estimated. We employed some traditional methods of source detection, for instance local peak detection, segmentation and thresholding. However, local peak detection was unable to perform a deblending, i.e, extended sources were considered as two separate objects. A deblending algorithm was applied where we successfully extracted all sources from images but was an inadequate method to pick only

extended sources from images. Next, the second method we applied, refers to the LULU and the Discrete Pulse Transform (DPT) algorithm. The algorithms aimed at removing impulse noise from an image and decomposed the latter into a collection of pulses where the noise and meaningful features were now separated. Then, a low and high pass filtering was applied with different thresholding. We noticed that with certain thresholds the algorithm successfully extracted only extended sources. However, this method was computationally expensive and hence not a feasible technique to apply when dealing with large volume of data. Finally, we applied a third method of extraction of sources using a filtering method. In this process, we employed the Otsu thresholding and Gaussian Filtering as preprocesses. Afterwards, a high-pass filtering algorithm was applied. We have therefore successfully extracted extended sources in an automated fashion.

Since machine learning (ML) classification algorithms generally learn from features extracted from the data, in Chapter 4 we introduced the concept of preprocessing and feature extraction. We applied this to the classification of Point-Extended sources and FRI-FRII sources. We also discussed the performance measures we utilize for assessing the classification algorithms - mainly accuracy, AUC, recall, precision and F1-score. Then, a description of the dataset was given where the latter was split into training and testing sets using k -Fold stratified cross-validation. Moreover, we carried out a shapelet transformation on the images to obtain sets of coefficients for each source. PCA was used as a dimensionality reduction technique since we were dealing with high dimensional coefficients. Finally, using 40 shapelets coefficients for each source, we employed the various ML algorithms for classification. The results of the classifications for the four classifiers were summarised. Using the shapelet coefficients as features, we observed that Point-Extended sources have been successfully classified with MLP being the best-performing classifier with an accuracy (89%), recall (84%), precision (96%), F1-score (88%) and AUC value (93%). The same procedures have been carried out for FRI-FRII sources. We noted that for these particular types of radio galaxies, the shapelet coefficients could not explain the distinct features between the two classes. We observed that RF is the best-performing algorithm with an accuracy (75%), recall (82%), precision (74%), F1-score (77%) and AUC value (74%) for FRI-FRII classification. The results obtained for FRI-FRII classification were less than manual classification. Since the shapelet transformation could not encapsulate the features of FRI and FRII sources, we adopted a technique employed in the field of computer vision for classification of images.

In Chapter 5, to perform the classification of FRI-FRII, we adopted a Deep Learning tech-

niques known as the Convolutional Neural Network (CNN) that directly learnt from the images instead of extracted features. To train the network, a large amount of data is required. In our case, since we have only a few training samples, we employed three methods of over-sampling, thus creating a large training set to optimally train the network. The first method involved the decomposition of an input image using shapelet transformation and reconstructed a model image by implementing the sampling method from a probability distribution using different σ s. The second method employed a rotation, horizontal and vertical reflection of input images. Finally, the third method involved the training of a Deep Convolutional Generative Adversarial Neural Network to generate examples of FRI and FR II images. We have adopted the DCGAN to train the CNN for the classification as in the parameter space, the layers will learn more evenly distributed samples. The classification of FRI-FR II radio galaxies are summarised with an accuracy (84%), AUC (85%), recall (84%), precision (83%), F1-score (83%). We have therefore presented the first application of DCGAN and CNN in radio astronomy and demonstrated that CNN together with DCGAN can accurately classify classes of radio galaxies. We have observed an increase of $\sim 9\%$ in accuracy with neural network compared to the ML algorithms applied in Chapter 4, hence CNN is a good approach for source classification.

Many possible avenues are opened for future work. We are currently working on an extension of this algorithm to be applied on sky images consisting of ~ 1000 point sources. The idea is to perform a classification of sources in a sky model without applying any cut-outs or having only one isolated source in an image. One main disadvantage of training deep learning models in astronomy is the availability of large sample datasets. Therefore, we basically want to apply a transfer learning approach to predict different radio galaxies using Deep Learning. The idea is to utilize pre-trained models: LeNet, AlexNet, VGG Net, ResNet and GoogLeNet trained on ImageNet. Features from the pre-trained network on ImageNet can be used for learning low-level and basic features like edges and bright spots. Since features from the pre-trained models are only object-centric images, we will have to employ a second transfer learning step where complicated features are utilized as a proxy for complex sources. The last few layers of the network learn complex features. We will therefore retrain only few last layers and the initial layers with the pre-trained weights will be kept.

Machine learning techniques are playing a role of ever-increasing importance in the field of radio astronomy. Artificial Intelligence (AI) is transforming society and with the petabytes of data to flow from future sky surveys, AI will also transform the way we do astronomy and science in general.

Appendix A

A Introduction to Bayes Theorem

In a machine learning perspective, we are usually interested in determining the *most probable* outcome or hypothesis from a space of all hypotheses Z , given the observed training data set D and any information about the prior probabilities of the different hypotheses. Bayes theorem allows a direct technique of calculating the probabilities of the different hypotheses in Z . More precisely, the probability of a particular hypothesis can be determined by using its prior probability and the probabilities of the observed training data set given the hypothesis and the observed data.

To introduce Bayes theorem in a mathematical formulation, some notation is introduced. The hypothesis space is denoted as $Z = \{z_1, z_2, \dots, z_m\}$, the *prior probability* of hypothesis z is represented by $P(z)$ which is independent of D , that is, some background information we have of z being the correct one. $P(D)$ is the probability of observing training data D given no information about the correct hypothesis. $P(D|z)$ is the probability that training data D will be observed given that hypothesis z holds. However, we are mostly interested in the *posterior probability* $P(z|D)$ that reflects our confidence that z holds given the data D . Bayes theorem defined in Equation A.1 is a cornerstone of the Bayesian learning approach and provides a technique for computing the posterior probability $P(z|D)$ from the prior probabilities of $P(z)$ and $P(D)$ as well as $P(D|z)$.

$$P(z|D) = \frac{P(D|z)P(z)}{P(D)} \quad (\text{A.1})$$

In many cases, it is of interest to find the most probable hypothesis $z \in Z$ if one considers some sample of hypotheses Z given the observed data D . The most probable hypothesis is known as a *maximum a posteriori* (MAP) hypothesis and this can be determined by taking the Bayes theorem to compute the posterior probability of individual hypothesis. Mathematically,

this can be written as in Equation A.2 where z_{MAP} is a MAP hypothesis if

$$z_{MAP} \equiv \operatorname{argmax}_{z \in Z} P(z|D)$$

$$z_{MAP} = \operatorname{argmax}_{z \in Z} \frac{P(D|z) P(z)}{P(D)}$$

The term $P(D)$ being a constant can be dropped to obtain:

$$z_{MAP} = \operatorname{argmax}_{z \in Z} P(D|z) P(z) \tag{A.2}$$

In some cases, when no prior information about any hypotheses is given, then we assume that all hypotheses have equal probability of being correct, that is, $P(z_i) = P(z_j)$ for all candidates in Z . Therefore, in such case, Equation A.2 can further be simplified by dropping the term $P(z)$, thus remaining with the only term $P(D|z)$ also known as the *likelihood* of D given z and the hypothesis that maximizes $P(D|z)$ is known as the *maximum likelihood* (ML) hypothesis, z_{ML} . Therefore, Equation A.2 can be written as

$$z_{ML} \equiv \operatorname{argmax}_{z \in Z} P(D|z) \tag{A.3}$$

Appendix B

B The algebraic derivation of PCA and the Maximum Variance Formulation

Consider $\{\mathbf{x}_n\}$ to be a data set of observations or set of images where $n = 1, 2, \dots, N$ and \mathbf{x}_n is a D -dimension Euclidean variable. The aim is to transform the data onto M -dimensional space where $M < D$, that allows maximizing the variance of the projected data. For now, assume the value of M is known and we begin the projection of the data onto a one-dimensional space where $M = 1$. A vector \mathbf{u}_1 of dimensionality- M is defined as the direction of the space and is chosen to be a unit vector as in (Equation B.1). Since here, we are only interested in the direction of \mathbf{u}_1 and not in its magnitude.

$$\mathbf{u}_1^T \mathbf{u}_1 = 1 \quad (\text{B.1})$$

Then, each data point \mathbf{x}_n is projected along \mathbf{u}_1 onto a scalar value as $\mathbf{u}_1^T \mathbf{x}_n$. The mean of the projected data $\mathbf{u}_1^T \bar{\mathbf{x}}$ with the data set mean $\bar{\mathbf{x}}$ is given by:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \quad (\text{B.2})$$

and the variance of the projected data is then

$$\frac{1}{N} \sum_{n=1}^N \left\{ \mathbf{u}_1^T \mathbf{x}_n - \mathbf{u}_1^T \bar{\mathbf{x}} \right\}^2 = \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 \quad (\text{B.3})$$

where \mathbf{S} is the covariance matrix of the data given by

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T \quad (\text{B.4})$$

In order to find the direction of projection \mathbf{u}_1 that would return most of the variance present in the original data, we maximize the projected variance $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$ in Equation B.3 with respect to \mathbf{u}_1 . In order to prevent $\|\mathbf{u}_1\| \rightarrow \infty$, this maximization has to be constrained and this can only be achieved by using the normalization condition given in Equation B.1. Therefore, a Lagrange multiplier λ_1 is introduced in order to enforce this constraint, given by:

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1) \quad (\text{B.5})$$

Setting the derivative of Equation B.5 with respect to \mathbf{u}_1 to zero, and transposing both sides, we find,

$$\mathbf{S} \mathbf{u}_1 - \lambda_1 \mathbf{u}_1 = 0 \quad (\text{B.6})$$

From Equation B.6, it is observed that \mathbf{u}_1 is an eigenvector of \mathbf{S} . Then, multiplying both sides with \mathbf{u}_1^T , we obtain

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1^T \mathbf{u}_1 \quad (\text{B.7})$$

Keeping in mind that $\mathbf{u}_1^T \mathbf{u}_1 = 1$, we have the following projected variance as:

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1 \quad (\text{B.8})$$

From Equation B.8, it is evident that in order to maximize the variance we have to set \mathbf{u}_1 to the eigenvector with the largest eigenvalue (λ_1). This eigenvector is also known as the *First Principal Component*. Additional PCs can be defined by each time choosing a new direction which maximizes the projected variance. Generally, for an M -dimensional principal subspace, the optimal linear projection that maximizes the projected variance is defined by the M eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_M$ of \mathbf{S} corresponding to the M largest eigenvalues $\lambda_1, \dots, \lambda_M$.

To summarize, PCA firstly calculates the mean $\bar{\mathbf{x}}$ and the covariance matrix \mathbf{S} of the data set. Then, the M eigenvectors of \mathbf{S} corresponding to the M largest eigenvalues are found. For feature extraction of an instance \mathbf{x}_n , M corresponding coefficients can then be utilized as the instance's new features. It is noted that \mathbf{x}_n undergoes dimensionality reduction as $M < D$. For a specific instance \mathbf{x}_n , the M -dimensional vector of coefficients, also known as PC Weights, is denoted by \mathbf{a}_n and is given as:

$$a_n = u^T(x_n - \bar{x})\mathbf{a}_n = \mathbf{U}^T(\mathbf{x}_n - \bar{\mathbf{x}}) \quad (\text{B.9})$$

where \mathbf{U} is a $D \times M$ dimensional matrix having the M largest eigenvectors (PCs) as columns. Note that \mathbf{a}_n is now the new feature vector of \mathbf{x}_n .

Appendix C

C Otsu Thresholding

Otsu's method is a type of binarization algorithm. It iterates through all the possible threshold values and calculates a variance (measure of spread) for the pixel levels on each side of t , i.e. the pixels that either fall in foreground or background. The objective of Otsu thresholding is to spot the threshold value, t where the sum of foreground and background spreads is at its minimum (Greensted 2010). The algorithm is explained using a simple 6x6 image as shown in Figure C.1.

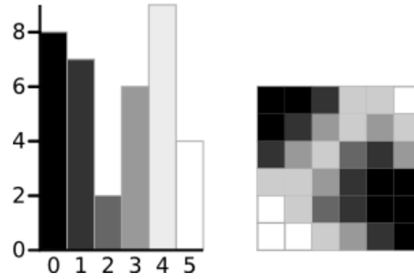


Figure C.1: The right panel shows a greyscale image of 6-levels and its histogram plot in the left panel. Figure adapted from (Greensted 2010)

In Otsu's algorithm, the threshold that minimizes the variance within the class, defined as a weighted sum of variances of the two classes given in Equation C.1 is searched exhaustively.

$$\text{Within Class Variance } \sigma_w^2(t) = \omega_b(t)\sigma_b^2(t) + \omega_f(t)\sigma_f^2(t) \quad (\text{C.1})$$

where the weights $\omega_{f,b}$ are the probabilities of the two classes separated by a threshold t and $\sigma_{f,b}$ are the variances of these two classes. The class probability $\omega_{f,b}(t)$ is computed from the histograms H .

$$\text{Weight for the background: } \omega_b(t) = \sum_{i=0}^{t-1} \mathcal{P}(i) \quad (\text{C.2})$$

$$\text{Weight for the foreground: } \omega_f(t) = \sum_{i=t}^{H-1} \mathcal{P}(i) \quad (\text{C.3})$$

$$\text{Mean for the background: } \mu_b(t) = \sum_{i=0}^{t-1} i \frac{\mathcal{P}(i)}{\omega_b} \quad (\text{C.4})$$

$$\text{Mean for the foreground: } \mu_f(t) = \sum_{i=t}^{H-1} i \frac{\mathcal{P}(i)}{\omega_f} \quad (\text{C.5})$$

where \mathcal{P} is the pixel values at different threshold. For a single threshold, $t = 3$, the calculations to obtain the foreground and background variances are shown.

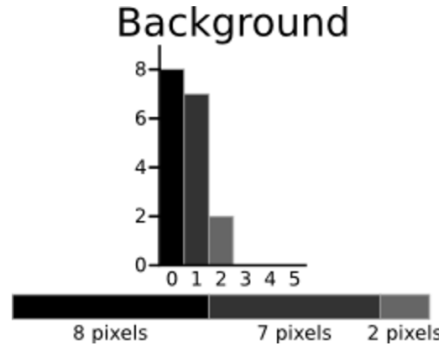


Figure C.2: Pixel values categorized as the background are less than a threshold 3. Figure adapted from (Greensted 2010)

$$\text{Weight, } \omega_b = \frac{8+7+2}{36} = 0.4722$$

$$\text{Mean, } \mu_b = \frac{(0 \times 8) + (1 \times 7) + (2 \times 2)}{17} = 0.6471$$

$$\text{Variance } \sigma_b^2 = \frac{((0-0.6471)^2 \times 8) + ((1-0.6471)^2 \times 7) + ((2-0.6471)^2 \times 2)}{17} = 0.4637$$

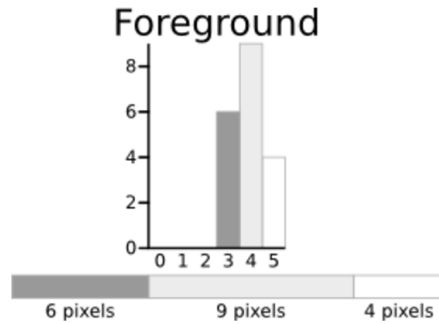


Figure C.3: Pixel values that are greater than a threshold 3 represent the foreground. Figure adapted from (Greensted 2010)

$$\text{Weight, } \omega_f = \frac{6+9+4}{36} = 0.5278$$

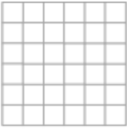
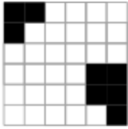
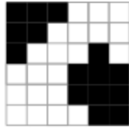
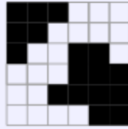
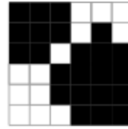
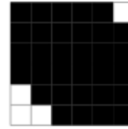
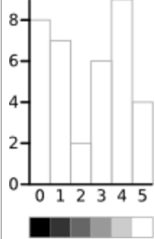
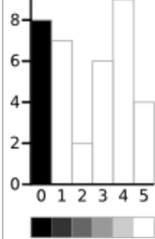
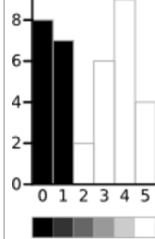
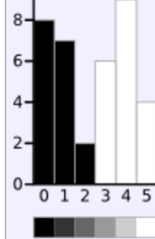
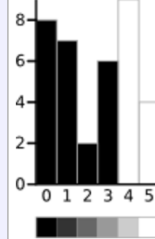
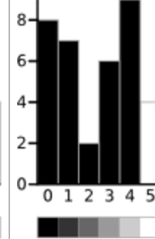
$$\text{Mean, } \mu_f = \frac{(3 \times 6) + (4 \times 9) + (5 \times 4)}{19} = 3.8947$$

$$\text{Variance } \sigma_f^2 = \frac{((3-3.8947)^2 \times 6) + ((4-3.8947)^2 \times 9) + ((5-3.8947)^2 \times 4)}{19} = 0.5152$$

Then using Equation C.1, the 'Within-Class Variance' for threshold, $t = 3$, is calculated by simply doing the sum of the two variances (foreground and background) multiplied by their associated weights.

$$\text{Within Class Variance, } \sigma_w^2 = (0.4722 \times 0.4637) + (0.5278 \times 0.5152) = 0.4909$$

Table C.1: The within class variance for all the possible thresholds. Figure adapted from (Greensted 2010)

| Threshold | T=0 | T=1 | T=2 | T=3 | T=4 | T=5 |
|------------------------------|---|---|---|--|---|---|
| |  |  |  |  |  |  |
| |  |  |  |  |  |  |
| Weight, Background | $W_b = 0$ | $W_b = 0.222$ | $W_b = 0.4167$ | $W_b = 0.4722$ | $W_b = 0.6389$ | $W_b = 0.8889$ |
| Mean, Background | $M_b = 0$ | $M_b = 0$ | $M_b = 0.4667$ | $M_b = 0.6471$ | $M_b = 1.2609$ | $M_b = 2.0313$ |
| Variance, Background | $\sigma_b^2 = 0$ | $\sigma_b^2 = 0$ | $\sigma_b^2 = 0.2489$ | $\sigma_b^2 = 0.4637$ | $\sigma_b^2 = 1.4102$ | $\sigma_b^2 = 2.5303$ |
| Weight, Foreground | $W_f = 1$ | $W_f = 0.7778$ | $W_f = 0.5833$ | $W_f = 0.5278$ | $W_f = 0.3611$ | $W_f = 0.1111$ |
| Mean, Foreground | $M_f = 2.3611$ | $M_f = 3.0357$ | $M_f = 3.7143$ | $M_f = 3.8947$ | $M_f = 4.3077$ | $M_f = 5.000$ |
| Variance, Foreground | $\sigma_f^2 = 3.1196$ | $\sigma_f^2 = 1.9639$ | $\sigma_f^2 = 0.7755$ | $\sigma_f^2 = 0.5152$ | $\sigma_f^2 = 0.2130$ | $\sigma_f^2 = 0$ |
| Within Class Variance | $\sigma_w^2 = 3.1196$ | $\sigma_w^2 = 1.5268$ | $\sigma_w^2 = 0.5561$ | $\sigma_w^2 = \mathbf{0.4909}$ | $\sigma_w^2 = 0.9779$ | $\sigma_w^2 = 2.2491$ |

Similar approach is performed for all the possible threshold values from 0 to 5. The Table C.1 shows the results of these calculations. The highlighted column shows that for $t = 3$, a lowest sum of weighted variance is obtained. Therefore, this is the final selected threshold. All pixels with a level less than 3 are background while all those with a level equal to or greater than 3 are foreground.

Bibliography

- Abadi, M. et al. (2015), 'TensorFlow: Large-scale machine learning on heterogeneous systems', URL <http://tensorflow.org/>.
- Andreon, S., Gargiulo, G., Longo, G., Tagliaferri, R. & Capuano, N. (2000), 'Wide field imaging - i. applications of neural networks to object detection and star/galaxy classification', *Monthly Notices of the Royal Astronomical Society* **319**, 700–716.
- Anguelov, B. (2008), 'Discrete pulse transform of images:algorithm and applications', *Proceeding International Conference Pattern Recongition* pp. 8–11.
- Anguelov, R. & Fabris-Rotelli, I. (2010), 'Lulu operators and discrete pulse transform for multidimensional arrays', *Image Processing, IEEE Transactions on* **19(11)**, 3012–3023.
- Aniyan, A. K. & Thorat, K. (2017), 'Classifying radio galaxies with convolutional neural network', *The Astrophysical Journal Supplement Series* **230**, 20.
- Ball, N. M., Brunner, R. J., Myers, A. D. & Tchong, D. (2006), 'Robust machine learning applied to astronomical data sets: Star-galaxy classification of the sloan digital sky survey dr3 using decision trees', *Astronomy and Astrophysics Journal* **650**, 497–509.
- Banfield, J. et al. (2015), 'Radio galaxy zoo: host galaxies and radio morphologies derived from visual inspection', *Monthly Notices of the Royal Astronomical Society* **453(3)**, 2326–2340.
- Barreiro, R. B., Sanz, J. L., Herranz, D. & Martinez-Gonzalez, E. (2003), 'Comparing filters for the detection of point sources', *MNRAS* **342**, 119.
- Bartelmann, M. & Schneider, P. (2008), 'Weak gravitational lensing', *Elsevier Journal* .
- Baum, S. A., Zirbel, E. L. & O'Dea, C. P. (1995), 'Toward understanding the fanaroff-riley dichotomy in radio source morphology and power', *Astronomy and Astrophysics Journal* **451**, 88.
- Bazell, D. & Peng, Y. (1998), 'A comparison of neural network algorithms and preprocessing methods for star-galaxy discrimination', *ApJS* **116**, 47–55.
- Becker, R. H., White, R. L. & Helfand, D. J. (1995), 'The first survey: The faint images of the radio sky at twenty centimeters', *The Astrophysical Journal* **450**, 559–577.
- Bentley, J. L. (1975), 'Multidimensional binary search trees used for associative searching', *Communications of the ACM* **18(9)**, 509–517.

- Bertin, E. & Arnouts, S. (1996), ‘SExtractor: Software for source extraction’, *Astronomy and Astrophysics Supplement series* **117**, 393–404.
- Bishop, C. (2000), *Pattern Recognition and Machine Learning*, Springer, New York.
- Booth, R., De Blok, W., Jonas, J. & Fanaroff, B. (2009), ‘Meerkat key project science, specifications, and proposals’, *arXiv:0910.2935*.
- Bradley, L., et al. (2016), ‘astropy/photutils: v0.3’, URL <http://doi.org/10.5281/zenodo.164986>.
- Bradt, H. (2004), ‘Astronomy methods: A physical approach to astronomical observations’, *Cambridge University Press, Cambridge, UK*.
- Breiman, L. (1994), ‘Bagging predictors’, *Technical report* **421**.
- Breiman, L. (2001), ‘Random forests’, *Machine Learning* **45**, 5–32.
- Brown, R. H. & Hazard, C. (1950), ‘Radio-frequency radiation from the great nebula in andromeda (m.31)’, *Nature* **166(4230)**, 901–902.
- Brown, R. H. & Hazard, C. (1951), ‘Radio emission from the andromeda nebula’, *MNRAS* **111**, 357–367.
- Buil, C. (1991), ‘Ccd astronomy : Ccd astronomy, construction and use of an astronomical ccd camera’, *William-Bell Inc. Virginia*.
- Butler-Yeoman, T., Frenn, M., Hollitt, C., Hogg, D. & Johnston-Hollitt, M. (2016), ‘Detecting diffuse sources in astronomical images’, *arXiv:1601.00266v1*.
- Capetti, A., Massaro, F. & Baldi, R. D. (2016), ‘Fricat: A first catalog of fr i radio galaxies’, *Astronomy and Astrophysics Journal* **598**, A49.
- Capetti, A., Massaro, F. & Baldi, R. D. (2017), ‘Friicat: A first catalog of fr ii radio galaxies’, *Astronomy and Astrophysics Journal* **302**A7.
- Carballo, R., Cofino, A. S. & Gonzalez-Serrano, J. I. (2004), ‘Selection of quasar candidates from combined radio and optical surveys using neural networks’, *MNRAS* **353**, 211–220.
- Carballo, R., Gonzalez-Serrano, J. I., Benn, C. R. & Jimenez-Lujan, F. (2008), ‘Use of neural networks for the identification of new $z > 3.6$ qsos from first-sdss dr5’, *MNRAS* **391**, 369–382.

- Chao, C., Liaw, A. & Breiman, L. (2004), 'Using random forests to learn imbalanced data', *University of California, Berkeley*.
- Chatfield, K., Simonyan, K., Vedaldi, A. & Zisserman, A. (2014), 'Return of the devil in the details: Delving deep into convolutional nets.', *CoRR*.
- Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002), 'Smote: Synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research* pp. 321–357.
- Chen, H. (1995), 'Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms', *Journal of the American society for Information Science* **46(3)**, 194.
- Chollet, F. (2016), 'Keras', URL <https://github.com/fchollet/keras>.
- Ciosa, K. J. & Moore, G. W. (2002), 'Uniqueness of medical data mining', *Artificial Intelligence in Medicine* **26(1)**, 1–24.
- Claeskens, J., Smette, A., Vandenbulcke, L. & Surdej, J. (2006), 'Identification and redshift determination of quasi-stellar objects with medium-band photometry: Application to gaia', *MNRAS* **367**, 879–904.
- Cohen, S. H., Windhorst, R. A., Odewahn, S. C., Chiarenza, C. A. & Driver, S. P. (2003), 'The hubble space telescope wfpc2 b-band parallel survey: A study of galaxy morphology', *AJ* **125**, 1762–1785.
- Colina, L. & Perez-Fournon, I. (1990), *Interaction Versus Radio Source Generation: A CCD Survey of Galaxies with Radio Jets*. In: Wielen R. (eds) *Dynamics and Interactions of Galaxies*, Springer, Berlin, Heidelberg.
- Condon, J. J., Cotton, W. D., Greisen, E. W. & Yin, Q. F. (1998), 'The nrao vla sky survey', *The Astrophysical Journal* **115**, 1693–1716.
- Connolly, A. J. & Szalay, A. S. (1999), 'A robust classification of galaxy spectra: Dealing with noisy and incomplete data', *AJ* **117**, 2052–2062.
- Connolly, A. J., Szalay, A. S., Bershad, M. A., Kinney, A. L. & Calzetti, D. (1995), 'Spectral classification of galaxies: an orthogonal approach', *AJ* **110**, 1071.
- Damiani, F., Maggio, A., Micela, G. & Sciortino, S. (1997), 'A method based on wavelet transforms for source detection in photon-counting detector images. i. theory and general properties', *MNRAS* **483**, 350–369.

- Danjuma, K. J. (2015), 'Performance evaluation of machine learning algorithms in post-operative life expectancy in the lung cancer patients', *CoRR* **abs/1504.04646**.
- Denton, E. L., Chintala, S., Szlam, A. & Fergus, R. (2015), 'Deep generative image models using a laplacian pyramid of adversarial networks', *CoRR* **abs/1506.05751**.
- Dieleman, S., Willett, K. W. & J., D. (2015), 'Rotation-invariant convolutional neural networks for galaxy morphology prediction', *MNRAS* **450**, 1441.
- Donoso, E. (2010), *Evolution of Radio Galaxies Across Cosmic Time*, PhD thesis, Ludwig Maximilian University of Munich.
- Dursun, D. (2009), 'Analysis of cancer data: a data mining approach', *Expert Systems: The Journal of Knowledge Engineering* **26(1)**, 100–112.
- Edwards, P. G. & Tingay, S. J. (2004), 'New candidate ghz peaked spectrum and compact steep spectrum sources', *Astronomy & Astrophysics Journal* **424**, 91–106.
- Eilek, J. A. (2014), 'The dynamic age of centaurus a', *New J. Phys* .
- Eisfeller, B. & Hein, G. (1994), 'Astrogeodetic levelling with an integrated dgps/ccd star camera system', *Proceedings of the International Symposium on Navigation (KISS 94) Calgary Canada* .
- Fabris-Rotelli, I. (2009), 'Lulu operators on multidimensional arrays and applications', *Masters dissertation, University of Pretoria* .
- Fabris-Rotelli, I. & Van der Walt, S. (2009), 'The discrete pulse transform in two dimensions', *Proceeding as part of the Twentieth Annual Symposium of the Pattern Recognition Association of South Africa* .
- Fanaroff, B. L. & Riley, J. M. (1974), 'The morphology of extragalactic radio sources of high and low luminosity.', *Monthly Notices Royal Astronomical Society* **167**, 31–36.
- Freeman, P. E., Kashyap, V., Rosner, R. & Lamb, D. Q. (2002), 'A wavelet-based algorithm for the spatial analysis of poisson data', *ApJS* **138**, 185–218.
- Friedman, J., Bentley, J. & Finkel, R. (1977), 'An algorithm for finding best matches in logarithmic expected time', *ACM Transactions on Mathematical Software* **3(3)**, 209–226.
- Gallagher, M. (1999), *Multi-layer Perceptron Error Surfaces: Visualization, Structure and Modelling*, PhD thesis, University of Queensland, Australia.

- Goldstein, D. A. e. a. (2015), ‘Automated transient identification in the dark energy survey’, *Astrophysical Journal* **150**, 82.
- Gonzalez, R. & Wintz, P. . (1987), ‘Digital image processing’, *Addison Welsley Publishing Co* .
- Goodfellow, I. et al. (2014), ‘Generative adversarial nets’, *Advances in Neural Information Processing Systems* pp. 2672–2680.
- Greensted, A. (2010), ‘Otsu thresholding’, URL <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>.
- Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2 edn, Springer, New York, NY.
- Heavens, A., Fantaye, Y., Sellentin, E., Eggers, H., Hosenie, Z., Kroon, S. & Mootooyaloo, A. (2017), ‘No evidence for extensions to the standard cosmological model’, *Physical Review Letters* **119(10-8)**, 101301.
- Herzog, A. D. & Illingworth, G. (1977), ‘The structure of globular clusters. i. direct plate automated reduction techniques’, *ApJS* **88**, 55–67.
- Hinton, G. et al. (2012), ‘Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups’, *IEEE Signal Processing Magazine* **29(6)**, 82–97.
- Hotelling, H. (1993), ‘Analysis of a complex of statistical variables into principal components’, *The Journal of Educational Psychology* **24**, 417–441.
- Hoyle, B. (2016), ‘Measuring photometric redshifts using galaxy images and deep neural networks’, *Astronomy and Computing* **16**, 34.
- Hubble, E. (1926), ‘Extragalactic nebulae’, *Astrophysical Journal* **64**, 321–369.
- Ian, H. W., Frank, E. & Hall, M. A. (2011), *Data Mining-Practical Machine Learning Tools and Techniques*, Burlington, MA USA: Morgan Kaufmann - Elsevier.
- Ioffe, S. & Szegedy, C. (2015a), ‘Batch normalization: Accelerating deep network training by reducing internal covariate shift.’, *arXiv:1502.03167* .
- Ioffe, S. & Szegedy, C. (2015b), ‘Batch normalization: Accelerating deep network training by reducing internal covariate shift’, *CoRR* **abs/1502.03167**.
- Irwin, M. J. (1985), ‘Automatic analysis of crowded fields’, *MNRAS* **214**, 575–604.

- Jansky, K. G. (1933), 'Electrical disturbances apparently of extraterrestrial origin', *Proceedings of the Institute of Radio Engineers* **21(10)**, 1387–1398.
- Jarvis, J. F. & Tyson, J. A. (1981), 'Focas: Faint object classification and analysis system', *The Astronomical Journal* **86 (3)**, 476–495.
- Jennison, R. & Das Gupta, M. (1953), 'Fine structure of the extra-terrestrial radio source cygnus 1', *Nature* pp. 996–997.
- Johnston, S., Taylor, R. & Bailes, M. (2008), 'Science with askap. the australian square-kilometre-array pathfinder', *Experimental Astronomy* **22(3)**, 151–273.
- Jolliffe, I. (2002), *Principal Component Analysis*, 2nd edn, Springer, New York.
- Khan, A. (2001), 'A regularization approach for variational inequalities', *Computers and Mathematics with Applications* **42(1-2)**, 65–74.
- Kharb, P., Stanley, E., Lister, M., Marshall, H., ODea, C. & Baum, S. (2015), 'Understanding jets from sources straddling the fanaroff-riley divide', *Proceedings IAU Symposium* **313**.
- Kim, E. J. & Brunner, R. J. (2016), 'Star-galaxy classification using deep convolutional neural networks', *MNRAS* **464**, 4463–4475.
- Kingma, D. P. & Ba, J. (2014), 'Adam: A method for stochastic optimization', *CoRR* **abs/1412.6980**.
- Kingma, D. P. & Ba, J. (2015), 'Adam: A method for stochastic optimization', *International Conference on Learning Representations* pp. 1–15.
- Knigge, C., Scaringi, S., Goad, M. R. & Cottis, C. E. (2008), 'The intrinsic fraction of broad-absorption line quasars', *MNRAS* **386**, 1426–1435.
- Konda, K., Bouthillier, X., Memisevic, R. & Vincent, P. (2015), 'Dropout as data augmentation', *arXiv:1506.08700*.
- Kremer, J., Stensbo-Smidt, K., Gieseke, F. & Igel, K. S. P. C. (2017), 'Big universe, big data: Machine learning and image analysis for astronomy', *IEEE Intelligent Systems* **32(3)**, 16 – 22.
- Krizhevsky, A., Sutskever, I. & Hinton, G. (2012), 'Imagenet classification with deep convolutional neural networks', *Advances in Neural Information Processing Systems* **25**.

- Kulkarni, V. & Sinha, P. (2013), 'Random forest classifiers: A survey and future research directions', *International Journal of Advanced Computing* **36(1)**, 1144–1153.
- Laney, D. (2001), 'Big data: The next frontier for innovation, competition, and productivity', *Application Delivery Strategies* **4**, 949.
- Lang, D., Hogg, D. W., Mierle, K., Blanton, M. & Roweis, S. (2010), 'Astrometry.net: Blind astrometric calibration of arbitrary astronomical images', *AJ* **139**, 1782–1800.
- Lawrence, S., Giles, C. L., Tsoi, A. C. & Back, A. D. (1997), 'Face recognition: A convolutional neural network approach', *IEEE Trans. on Neural Networks* **8(1)**, 98.
- Lazio, T. J. W. (2009), 'The square kilometre array', *Proceedings of Science* .
- Lazzati, D., Campana, S., Rosati, P., Panzera, M. R. & Tagliaferri, G. (1999), 'The brera multi-scale wavelet (bmw) rosat hri source catalog i: the algorithm', *ApJ* **524**, 414–422.
- LeCun, Y. & Bengio, Y. (1995), *Convolutional networks for images, speech, and time-series*, MIT Press.
- LeCun, Y., Bengio, Y. & Hinton, G. (2011), 'Deep learning', *Nature* **521**, 436–444.
- LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998), 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE* **86**, 2278–2324.
- Leedham, G., Yan, C., Takru, K., Nata Tan, J. H. & Mian, L. (2003), 'Comparison of some thresholding algorithms for text/background segmentation in difficult document image', *Proceedings of the Seventh International Conference on Document Analysis and Recognition* **2**, 859.
- Lord Rayleigh, F. (1879), 'Investigations in optics, with special reference to the spectroscope', *Philosophical Magazine* **8(49)**, 261–274.
- Lynden-Bell, D. (1969), 'Galactic nuclei as collapsed old quasars', *Nature* **223 (5207)**, 690–694.
- Maas, A. L., Hannun, A. Y. & Ng, A. Y. (2013), Rectifier nonlinearities improve neural network acoustic models.
- Madgwick, D., Lahav, O., Taylor, K. & 2dFGRS Team (2001), 'Parameterisation of galaxy spectra in the 2df galaxy redshift survey', *Mining the Sky* p. 331.
- Madgwick, D. S. (2003), 'Correlating galaxy morphologies and spectra in the 2df galaxy redshift survey', *Monthly Notices of the Royal Astronomical Society* **338**, 197–207.
- Makovoz, D. & Marleau, F. R. (2006), 'Point-source extraction with mopex', *PASP* **459**, 341–352.

- Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. & Byers, A. (2011), 'Big data: The next frontier for innovation, competition, and productivity', *McKinsey Global Institute* .
- Mao, S. (1999), 'Procs of gravitational lensing: Recent progress and future goals', *Boston University* .
- Mauch, T., Murphy, T., Buttery, H., Curran, J., Hunstead, R., Piestrzynski, B., Robertson, J. & Sadler, E. (2003), 'Sumss: A widefield radio imaging survey of the southern sky ii. the source catalogue', *Royal Astronomical Society* **000**, 1–15.
- Mitchell, T. M. (1997), *Machine Learning*, McGraw-Hill, New York, NY, international edition.
- Nair, V. & Hinton, G. (2010a), 'Rectified linear units improve restricted boltzmann machines', *Proceedings of the 27th International Conference on Machine Learning* .
- Nair, V. & Hinton, G. E. (2010b), 'Rectified linear units improve restricted boltzmann machines', *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* **115**, 807–814.
- Newell, B. & O'Neil, E. J. (1977), 'The reduction of panoramic photometry. i. two search algorithms', *PASP* **89**, 925–928.
- Norris, R. P. et al. (2011), 'Emu: Evolutionary map of the universe', *Publications of the Astronomical Society of Australia* **23(3)**, 215–248.
- Ocana, F. B., Leona, S., Limb, J., Combesc, F. & Dinh-V-Trungb (2008), 'Molecular gas in nearby elliptical radio galaxies', *AIP Conference Proceedings* **1035**, 132.
- Odewahn, S. C. & Nielsen, M. L. (1994), 'Star-galaxy separation using neural networks', *Vistas in Astronomy* **38**, 281–286.
- Odewahn, S. C., Stockwell, E. B., Pennington, R. L., Humphreys, R. M., & Zumach, W. A. (1992), 'Automated star/galaxy discrimination with neural networks', *Astrophysics Journal* **103**, 318–331.
- Odewahn, S. C., Windhorst, R. A., Driver, S. P. & Keel, W. C. (1996), 'Automated morphological classification in deep hubble space telescope ubvi fields: Rapidly and passively evolving faint galaxy populations', *ApJL* **472**, L13–L16.
- Odewahn, S. C. et al. (2004), 'The digitized second palomar observatory sky survey (dposs). iii. star-galaxy separation', *Astrophysics Journal* **128**, 3092–3107.

- Ossama, A., Abdel-rahman, M., Hui, J., Li, D., Gerald, P. & Dong, Y. (2014), 'Convolutional neural networks for speech recognition', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **22(10)**, 1533–1545.
- Otsu, N. (1979), 'A threshold selection method from gray-level histograms', *In Trans. SMC* **9**, 62–66.
- Pearson, K. (1901), 'On lines and plates of closest fit to systems of points in space', *The London Edinburg and Dublin Philosophical Magazine and Journal of Science* **6th Series**, 559–572.
- Pedregosa, F. et al. (2011), 'Scikit-learn: Machine learning in python', *Journal of Machine Learning Research* **12**, 2825–2830.
- Perlmutter, S. et al. (1995), 'Cosmology from type ia supernovae', *LBNL* .
- Perret, B., Lefevre, S. & Collet, C. (2008), 'A robust hit-or-miss transform for template matching applied to very noisy astronomical images', *Pattern Recognition* **42**, 470–2480.
- Perucho, M., Marti, J. M., Laing, R. A. & Hardee, P. E. (2014), 'On the deceleration of far-off?riley class i jets: mass loading by stellar winds', *MNRAS* **441**, 1488–1503.
- Perucho, M. & Marti, M. J. (2007), 'A numerical simulation of the evolution and fate of a fri jet. the case of 3c 31', *MNRAS* pp. 1–19.
- Philip, N. S., Wadadekar, Y., Kembhavi, A. & Joseph, K. B. (2002), 'A difference boosting neural network for automated star-galaxy classification', *Astronomy and Astrophysics Journal* **385**, 1119–1126.
- Polsterer, K. L., Gieseke, F. C. & Igel, C. (2015), 'Automatic galaxy classification via machine learning techniques: Parallelized rotation/flipping invariant kohonen maps (pink)', *Astronomical Society of the Pacific* **495(24)**, 81–86.
- Proctor, D. D. (2016), 'A selection of giant radio sources from nvss', *The Astrophysical Journal Supplement Series* **224(2)**.
- Radford, A., Metz, L. & Chintala, S. (2015), 'Unsupervised representation learning with deep convolutional generative adversarial networks', *ArXiv e-prints* pp. 559–577.
- Rahmat, R., Malik, A. S. & Kamel, N. (2013), 'Comparison of lulu and median filter for image denoising', *International Journal of Computer and Electrical Engineering* **5(6)**.

- Refaeilzadeh, P., Tang, L. & Liu, H. (2009), 'Cross-validation', *In Encyclopedia of database systems* .
- Refregier, A. (2008), 'Shapelets i: A method for image analysis', *Royal Astronomical Society* **000**, 1–13.
- Ren, J., Lee, S. D., Chen, X., Kao, B., Cheng, R. & Cheung, D. (2009), 'Naive bayes classification of uncertain data', *Data Mining, Ninth IEEE International Conference* pp. 1550–4786.
- Riess, A. (1998), 'X-ray emmissions from clusters of galaxies', *Cambridge University Press: Cambridge* .
- Riggi, S., Ingallinera, A., Leto, P., Cavallaro, F., Bufano, F., Schilliro, F., Trigilio, C., Umana, G., Buemi, C. S. & Norris, R. P. (2016), 'Automated detection of extended sources in radio maps: progress from the scorpio survey', *Monthly Notices of the Royal Astronomical Society* **460**, 1486–1499.
- Rohwer, C. (2005), 'Nonlinear smoothers and multiresolution analysis', *Birkhauser* .
- Rokach, L. & Maimon, O. (2008), 'Data mining with decision trees: theory and applications', *World Scientific Pub Co Inc* .
- Schwinger, J. (1949), 'On the classical radiation of accelerated electrons', *Physical Review* **75**, 1912–1925.
- Senthilkumaran, N. & Vaithegi, S. (2016), 'Image segmentation by using thresholding techniques for medical images', *Computer Science and Engineering: An International Journal (CSEIJ)* **6(1)**.
- Seyfert, C. K. (1943), 'Nuclear emission in spiral nebulae', *Astrophysical Journal* **97**, 28–40.
- Shklovsky, I. (1958), 'On the nature of the emission from the galaxy ngc 4486', *In IAU Symp. 6: Electromagnetic Phenomena in Cosmical Physics* p. 517.
- Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W. & Webb, R. (2016), 'Learning from simulated and unsupervised images through adversarial training.', *arXiv:1612.07828* .
- Simard, P. Y., Steinkraus, D. & Platt, J. C. (2003), 'Best practices for convolutional neural networks applied to visual document analysis.', *ICDAR* **3**, 958–962.
- Slezak, E., Bijaoui, A. & Mars, G. (1999), 'Galaxy counts in the coma supercluster field', *ApJ* **524**, 414–422.

- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014), ‘Dropout: a simple way to prevent neural networks from overfitting.’, *Journal of Machine Learning Research* **15**(1), 1929–1958.
- Starck, J. & Bobin, J. (2010), ‘Astronomical data analysis and sparsity: From wavelets to compressed sensing’, *Proc. IEEE*. **98**, 1021.
- Stetson, P. B. (1987), ‘Daophot. a computer program for crowded-field stellar photometry’, *PASP* **99**, 191–222.
- Stoltz, G. & Fabris-Rotelli, I. (2015), ‘Pulse reformation algorithm for leakage of connected operators’, *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* **1**, 583–590.
- Storrie-Lombardi, M. C., Lahav, O., L. Sodr, J. & Storrie-Lombardi, L. J. (1992), ‘Morphological classification of galaxies by artificial neural networks’, *Monthly Notices of the Royal Astronomical Society* **259**, 8–12.
- Strack, J., Murtagh, F. & Bijaoui, A. (1998), ‘Image processing and data analysis’, *Cambridge: Cambridge, UK*, .
- Suszynski, R. & Wawryn, K. (2015), ‘Stars centroid determination using psf-fitting method’, *Metrology and Measurement Systems* **4**, 547–558.
- Szegedy, C., Toshev, A. & Erhan, D. (2013), ‘Deep neural networks for object detection’, *NIPS* .
- Urry, C. M. & Padovani, P. (1995), ‘Unified schemes for radio-loud active galactic nuclei’, *Publications of the Astronomical Society of the Pacific* **107**, 803.
- Van der Walt, S. et al. (2014), ‘scikit-image: image processing in Python’, *PeerJ* **2**, e453.
- Van Velzen, S., Falcke, H. & Kording, E. (2015), ‘VizieR online data catalog: Double-lobed radio sources catalog’, *Monthly Notices of the Royal Astronomical Society* **446**, 2985.
- Vdumoulin (2016), ‘A technical report on convolution arithmetic in the context of deep learning’, URL https://github.com/vdumoulin/conv_arithmetic.
- Vikhlinin, A., Forman, W., Jones, C. & Murray, S. (1995), ‘Matched filter source detection applied to the rosat pspc and the determination of the number-flux relation’, *ApJ* **451**, 542–552.
- Vinyals, O., Toshev, A., Bengio, S. & Erhan, D. (2015), ‘Show and tell: A neural image caption generator’, *Proceedings of the IEEE* .

- Vyas, A., Roopashree, M. & Prasad, B. (2010), 'Cetroid detection by gaussian pattern matching in adaptive optics', *International Journal of Computer Applications* **26(1)**, 30–36.
- Waisberg, I. R. (2013), 'Astronomical point source classification through machine learning', CS 229 *Machine Learning Final Projects* .
- Weir, N., Fayyad, U. M. & Djorgovski, S. (1995), 'Automated star/galaxy classification for digitized poss-ii', *Astrophysics Journal* **109**, 2401.
- White, R. L. (2000), 'The first bright quasar survey. ii. 60 nights and 1200 spectra later', *ApJS* **126**, 133–207.
- White, R. L., Becker, R. H., Helfand, D. J. & Gregg, M. D. (1997), 'A catalog of 1.4 ghz radio sources from the first survey', *The Astrophysical Journal* **475**, 479–493.
- Windhorst, R., Odewahn, S., Burg, C., Cohen, S., & Waddington, I. (1999), 'Young and old galaxies at high redshift', *ApSS* **269**, 243–262.
- Xu, B., Wang, N., Chen, T. & Li, M. (2015), 'Empirical evaluation of rectified activations in convolutional network', *CoRR* **abs/1505.00853**.
- Yang, Y., Li, N. & Zhang, Y. (2008), 'Automatic moving object detecting tracking from astronomical ccd image sequences', *In Proc. ICSMC* pp. 650–655.
- Yeh, R., Chen, C., Lim, T., Hasegawa-Johnson, M. & Do, M. N. (2016), 'Semantic image inpainting with perceptual and contextual losses', *CoRR* .
- Yip, C. W., Connolly, A. J., Szalay, A. S., Budavari, T., SubbaRao, M., Frieman, J. A., Nichol, R. C. & Hopkins, A. M. (2004), 'Distributions of galaxy spectral types in the sloan digital sky survey', *AJ* **128**, 585–609.
- Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014), 'How transferable are features in deep neural networks?', *Advances in Neural Information Processing Systems* pp. 3320–3328.
- Zhang, Y. & Zhao, Y. (2007), 'A comparison of bbn, adtree and mlp in separating quasars from large survey catalogues', *Chinese Journal of Astronomy and Astrophysics* **7**, 289.
- Zirbel, E. L. (1996), 'Host galaxy sizes of powerful radio sources', *Astrophysical Journal* **473**, 144.