

---

# Prospecting for enigmatic radio sources with autoencoders: a novel approach

---

By

Fernando Louis VENTURA



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Department of Physics  
UNIVERSITY OF PRETORIA

Submitted in partial fulfilment of the requirements for the  
degree of MASTER OF SCIENCE (MSC) IN PHYSICS in the  
Faculty of Natural and Agricultural Sciences.

August 3, 2022

*Supervisors:* Prof. Roger DEANE, A/Prof. Christopher CLEGHORN and Dr. Kshitij  
THORAT

UNIVERSITY OF PRETORIA

*Abstract*Faculty of Natural and Agricultural Sciences  
Department of Physics

Master of Science (MSc) in Physics

**Prospecting for enigmatic radio sources with autoencoders: a novel approach**

by Fernando Louis VENTURA

Supervisors: Prof. Roger DEANE, A/Prof. Christopher CLEGHORN and Dr. Kshitij THORAT


Modern and future radio surveys performed with increasingly powerful instruments, such as the 64-antenna MeerKAT interferometer and eventually the Square Kilometre Array (SKA), will catalogue upwards of hundreds of thousands to millions of radio sources. This can make classification of source morphology and searching for specific source classes extremely challenging. MeerKAT excels at imaging large-scale and faint emission features due to its high sensitivity and excellent imaging quality, allowing for many exotic, scientifically-rich radio objects to be identified for the first time. However, finding them is a problem, especially using manual classification. Moreover, MeerKAT's moderate angular resolution ( $\sim 5$  arcsec) means that a typical field is crowded with many sources, including many point-like sources.

An automated approach to classification is therefore required. The aim of this project is to isolate the most morphologically unusual or exotic sources. The approach explored in this project is the use of autoencoders, neural networks that encode an input into some latent space and then attempt to reconstruct the input from the code form.

We test this on the MeerKAT Galaxy Cluster Legacy Survey, comprising of 115 galaxy clusters at 1.28 GHz with  $\mu\text{Jy}/\text{beam}$  sensitivity. A subset of these are manually classified and used to train numerous configurations of autoencoder algorithms, including ensembles of autoencoders, and test the algorithms' performance in isolating potentially interesting sources. It is found that the autoencoders significantly reduce the work required to locate potentially interesting sources.

## Declaration of Authorship

I, Fernando Louis VENTURA, declare that the thesis, which I hereby submit for the degree of MSc in Physics at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

Signature: 

---

Date: 03-08-2022

---

## *Acknowledgements*

I would like to thank my supervisor, Roger Deane, and co-supervisors, Kshitij Thorat and Christopher Cleghorn. Their guidance and expertise was invaluable in the completion of this project. I would also like to thank the other staff, researchers and students in the Astronomy group and Department of Physics at the University of Pretoria. We acknowledge the use of the ilifu cloud computing facility – [www.ilifu.ac.za](http://www.ilifu.ac.za), a partnership between the University of Cape Town, the University of the Western Cape, the University of Stellenbosch, Sol Plaatje University, the Cape Peninsula University of Technology and the South African Radio Astronomy Observatory. The Ilifu facility is supported by contributions from the Inter-University Institute for Data Intensive Astronomy (IDIA – a partnership between the University of Cape Town, the University of Pretoria, the University of the Western Cape and the South African Radio astronomy Observatory), the Computational Biology division at UCT and the Data Intensive Research Initiative of South Africa (DIRISA).

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Declaration of Authorship</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>1 Objectives and Layout</b>	<b>1</b>
1.1 Thesis Objectives . . . . .	1
1.2 Thesis Layout . . . . .	1
<b>2 Introduction</b>	<b>3</b>
2.1 Radio Sources . . . . .	3
2.2 Radio Galaxy Morphologies . . . . .	4
2.3 The MeerKAT Galaxy Cluster Legacy Survey . . . . .	9
2.4 Neural Networks . . . . .	13
2.5 Convolutional Neural Networks . . . . .	15
2.6 Training . . . . .	16
2.7 Pooling Layers . . . . .	18
2.8 Autoencoders . . . . .	20
2.9 Performance Evaluation . . . . .	22
2.10 Ensembles . . . . .	24
2.11 Current Approaches to Classification and Anomaly Detection . . . . .	26
<b>3 Methods</b>	<b>32</b>
3.1 Data Processing . . . . .	32
3.2 Autoencoders . . . . .	36
3.3 Ensembles . . . . .	36
3.4 Performance Evaluation . . . . .	39
<b>4 Results</b>	<b>43</b>
4.1 Single Autoencoder . . . . .	43
4.2 Ensemble of Three Similar Autoencoders . . . . .	47
4.3 Ensemble of Three Different Autoencoders . . . . .	49
4.4 Ensemble of Nine Different Autoencoders . . . . .	57
4.5 A Sample of Automatically Selected Interesting Sources . . . . .	60
<b>5 Conclusions</b>	<b>62</b>
<b>Bibliography</b>	<b>64</b>

# List of Figures

2.1	Reproduced from Condon et al. (2012), brightness-weighted counts of 1.4 GHz sources as a function of source contribution to sky background temperature, showing differences between star formation and AGN driven sources. . . . .	4
2.2	Unified AGN model showing SMBH, accretion disk, torus and jets. . . . .	5
2.3	Typical examples of FRI and FR II radio galaxies. . . . .	6
2.4	Typical examples of FRI and FR II radio galaxies from the MGCLS data. . . . .	6
2.5	Reproduced from Owen and Ledlow (1994), the radio luminosity compared to the isophotal magnitude for a selection of FRI and FR II radio galaxies. . . . .	7
2.6	Reproduced from Zirbel and Baum (1994), the correlation between radio core power and emission line luminosities compared for FRI and FR II galaxies. . . . .	8
2.7	Reproduced from Leahy and Williams (1984), diagrams showing three mechanisms of radio galaxy jet distortions by their environment. . . . .	10
2.8	Reproduced from Cotton et al. (2020), an example of an X-shaped radio galaxy created by hydrodynamical backflow. . . . .	11
2.9	Examples of exotic radio galaxies from MGCLS. . . . .	12
2.10	The layout of the MeerKAT antennas. . . . .	12
2.11	Reproduced from Knowles et al. (2022), the capabilities of the MGCLS data is shown through some examples. . . . .	13
2.12	Reproduced from Nielsen (2015), A basic neural network example. . . . .	14
2.13	An illustration of how a max pooling operation works. . . . .	19
2.14	Reproduced from He et al. (2014), an illustration of how spatial pyramid pooling works. . . . .	21
2.15	Diagram of the basic structure of an autoencoder. . . . .	22
2.16	The confusion matrix for a binary classification. . . . .	23
2.17	Two figures reproduced from Fawcett (2006), illustrations of where various classifiers would fall on an ROC graph according to their performance and examples of the AUC. . . . .	25
2.18	Three figures reproduced from Mostert et al. (2021) showing a trained SOM's performance in classifying various types of sources. . . . .	30
2.19	Reproduced from Lochner and Bassett (2021), the results of Astronomy using Galaxy Zoo data. . . . .	31
3.1	An example of augmentations applied to an image. . . . .	34
3.2	Diagram of the process of making cutouts of the sources in the MGCLS images. . . . .	35
3.3	Diagram of an autoencoder of the type that is used in this thesis, showing the various layers. . . . .	36
3.4	Diagram showing three typical autoencoders working together in ensemble. . . . .	38

3.5	Summary of the methods for determining the classification of a single image using a given threshold. . . . .	41
4.1	PCA output for the single autoencoder when run on the encoded output in latent space. . . . .	43
4.2	Instance of the results produced by the single autoencoder, showing the distributions of NCC values for typical and exotic sources. . . . .	44
4.3	Ten best and worst reconstructed sources from both exotic and typical test sources produced by an instance of the single autoencoder, showing the original image, reconstruction and residual for each. . . . .	45
4.4	ROC curve for the single autoencoder. . . . .	46
4.5	Confusion matrices optimised for $F_1$ and $F_2$ scores for the single autoencoder. . . . .	47
4.6	Instance of the results produced by an ensemble of three similar autoencoders, showing the distributions of averaged NCC values for typical and exotic sources. . . . .	47
4.7	ROC curves for the ensemble of three similar autoencoders for averaging and voting methods. . . . .	48
4.8	Confusion matrices optimised for $F_1$ and $F_2$ scores using the averaging method for the ensemble of three similar autoencoders. . . . .	49
4.9	Confusion matrices optimised for $F_1$ and $F_2$ scores using the voting method for the ensemble of three similar autoencoders. . . . .	49
4.10	PCA outputs for the three differing autoencoders when run on the encoded outputs in latent space. . . . .	50
4.11	Instance of the results produced by an ensemble of three differing autoencoders, showing the distributions of averaged NCC values for typical and exotic sources. . . . .	50
4.12	Ten best and worst reconstructed exotic and typical test sources for an instance of the $16 \times 16 \times 16$ autoencoder from the three differing autoencoders, showing original image, reconstruction and residual. . . . .	52
4.13	Ten best and worst reconstructed exotic and typical test sources for an instance of the $8 \times 8 \times 32$ autoencoder from the three differing autoencoders, showing original image, reconstruction and residual. . . . .	53
4.14	Ten best and worst reconstructed exotic and typical test sources for an instance of the $32 \times 32 \times 32$ autoencoder from the three differing autoencoders, showing original image, reconstruction and residual. . . . .	54
4.15	ROC curves for the ensemble of three differing autoencoders for averaging and voting methods. . . . .	55
4.16	Confusion matrices optimised for $F_1$ and $F_2$ scores using the averaging method for the ensemble of three differing autoencoders. . . . .	55
4.17	Confusion matrices optimised for $F_1$ and $F_2$ scores using the voting method for the ensemble of three differing autoencoders. . . . .	56
4.18	Instance of the results produced by an ensemble of nine autoencoders, showing the distributions of averaged NCC values for typical and exotic sources. . . . .	57
4.19	ROC curves for the ensemble of nine autoencoders for averaging and voting methods. . . . .	58
4.20	Confusion matrices optimised for $F_1$ and $F_2$ scores using the averaging method for the ensemble of nine autoencoders. . . . .	58
4.21	Confusion matrices optimised for $F_1$ and $F_2$ scores using the voting method for the ensemble of nine autoencoders. . . . .	59

4.22 Manual selection of some testing images marked as exotic by the autoencoders. . . . . 61



# List of Tables

3.1 Table of the various ensembles used in this thesis. . . . .	38
---	----

## Chapter 1

# Objectives and Layout

### 1.1 Thesis Objectives

The aim of this project is to identify unusual radio sources in a subset of the MeerKAT Galaxy Cluster Legacy Survey (MGCLS, Knowles et al. 2022) using autoencoders, an automated unsupervised machine learning approach. MeerKAT, originally named the Karoo Array Telescope (KAT) before being renamed to MeerKAT (Afrikaans for "More of KAT"), is the 64 antenna radio telescope that is a precursor to the Square Kilometre Array (SKA) telescope and which will eventually form part of the mid-frequency component of the SKA. For this task, various ensembles of autoencoders are tried and compared. An automated approach for finding interesting objects in surveys is desirable due to the large number of sources that can be observed by modern surveys, which can typically exceed several hundreds of thousands, coupled with the dramatic increase in recovered source complexity with advanced interferometry. As new telescopes such as MeerKAT, and eventually the SKA, have high sensitivity and moderate resolution ( $\sim 5$  arcsec in the case of MeerKAT), diffuse sources may be observed but the fields are often crowded and complex. The use of machine learning is promising here as it may learn to deal with the unique and unusual features and non-uniform background noise of various sources that belong to the same category.

The objective is to explore if autoencoders show potential in finding exotic radio sources by morphology in survey data. The goal is to have practical software that may be used to reduce the time and effort required to search for individual sources that may be of interest to a researcher in the large surveys produced by modern and upcoming telescopes. By reducing the number of sources that the researcher needs to manually inspect while at the same time retaining a majority of the interesting sources, it may take less time or fewer experts to search the data for interesting sources for future study. The scope does not extend to the application of automatically classifying large numbers of typical sources in a reliable manner or collecting reliable statistical data about the morphologies of the sources in such surveys.

### 1.2 Thesis Layout

In the first part of the thesis we discuss the various types of sources that we seek to identify as well as the particular survey that they come from. The machine learning approach used is then discussed, as well as the data pre-processing of the survey images. The performance evaluation methods used to determine the effectiveness

of the various machine learning approaches is also discussed. Finally, the results of the various machine learning approaches are presented.

The thesis is divided into the following parts:

- **Chapter 1: Objectives and Layout**
- **Chapter 2: Introduction** In the introduction many of the important ideas are introduced including the survey data used to search for sources in this project, the software components used in the design of the autoencoders and how the performance of the autoencoders may be evaluated and understood.
- **Chapter 3: Methods** This chapter goes over the specific methods and software used for pre-processing the data, constructing and training the autoencoders as well as the evaluation of the performance of the autoencoders.
- **Chapter 4: Results** This chapter contains the results of the various configurations of autoencoder ensembles implemented as discussed in Chapter 2.
- **Chapter 5: Conclusions**

## Chapter 2

# Introduction

### 2.1 Radio Sources

First we must consider the types of sources that will be found in the data. The majority of the spatially resolved sources found in this survey and of interest to this project are radio galaxies. This is as radio galaxies are among the brightest radio sources ( $> \sim 1$  mJy) (where Jy is a unit of spectral flux density, equal to  $10^{-26} \text{ W m}^{-2} \text{ Hz}^{-1}$ ), dominating the high end of the radio luminosity function. For example, Figure 2.1, reproduced from Condon et al. (2012), shows the flux-density of various published 1.4 GHz source counts and shows the difference between the Active Galactic Nuclei (AGN) and star formation powered sources. Radio galaxies have an AGN at their centre of the galaxy which is the central engine of their radio emission. AGN consist of a central supermassive black hole (SMBH), with masses typically in the range of millions to billions of solar masses. The SMBH is actively accreting matter forming a thin disk about the SMBH. At the edge of this accretion disk a thick torus of gas and dust accumulates. Not all of the matter from the disk eventually falls into the black hole, however. Much of the matter is flung out by the AGN in large jets at relativistic velocities typically perpendicular to the disk of matter being accreted. This is likely due to the twisted magnetic fields threading through the accretion disk and possibly the SMBH itself (Blandford, Meier, and Readhead 2019, and references therein). These jets typically extend well beyond the host galaxy for hundreds of kpc (kiloparsec where a parsec is  $3.0857 \times 10^{16}$  m). Figure 2.2 shows an illustration of the AGN with its accretion disk, torus and jets. Not all supermassive black holes at the centre of galaxies accrete matter at a rate sufficient for a detectable disk (Fraknoi et al. 2016).

Within the accretion disk the plasma orbits the black hole at relativistic velocities. Viscosity-driven angular momentum loss, the result of viscous friction forces between gas orbiting at different speeds throughout the disk, converts gravitational potential energy to kinetic energy, driving high temperatures, typically on the order of  $10^6$ – $10^8$  K (Fraknoi et al. 2016). As a result, they emit strong blackbody radiation (thermal electromagnetic radiation emitted by matter with a temperature above absolute zero and with wavelength spectrum dependent only on temperature) with a peak that can lie in the ultraviolet or X-ray part of the spectrum.

Within the jets of matter ejected by the AGN, the charged particles, such as electrons, will spiral around the strong magnetic fields they propagate, producing synchrotron radiation (electromagnetic radiation emitted by radially accelerated charged particles) (Eilek 2014). These jets are bright in the radio part of the spectrum, and extend far beyond the borders of their host galaxies for the most radio-luminous systems (Fraknoi et al. 2016). They are the most prominent feature of radio galaxies

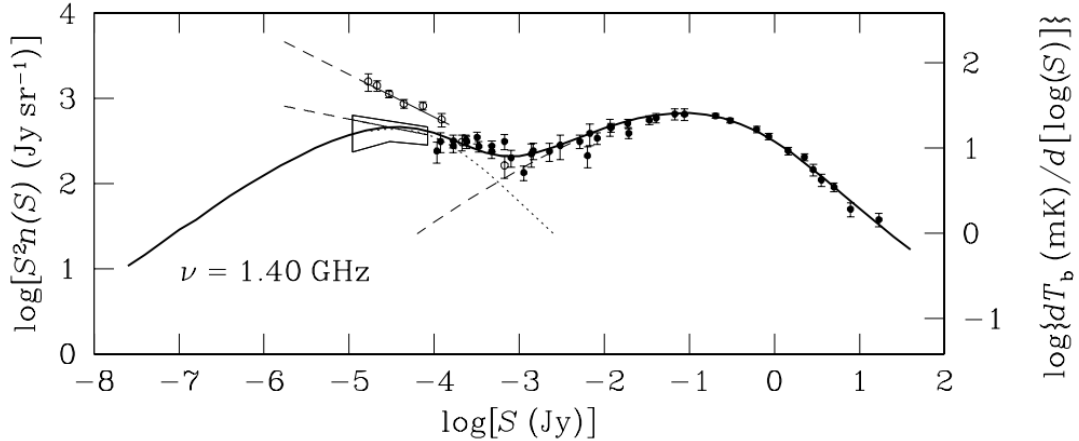


FIGURE 2.1: Reproduced from Condon et al. (2012). For published 1.4 GHz source counts the log of the brightness-weighted count,  $S^2n$ , is plotted as a function of  $\log(S)$ . The brightness-weighted source count is proportional to the flux density contribution per decade to the sky background temperature,  $T_b$ . The left ordinate is the log of the 1.4 GHz source count,  $S^2n$ , and right ordinate is the log of the source contribution per decade of flux density to the 1.4 GHz background,  $dT_b/d[\log(S)]$ . The solid curve shows the Condon (1984a) model count composed primarily of star formation, indicated with the dotted curve, and AGN driven sources, indicated with the dashed curve. Filled data points at  $\log[S(\text{Jy})] > -3$  are from Condon (1984b) and Mitchell and Condon (1985). Open data points are from Owen and Morrison (2008), source count with their power law fit.

when observed using a radio telescope due to both their projected size and brightness.

## 2.2 Radio Galaxy Morphologies

Now we consider the way in which these sources differ from each other and how they may be classified according to their morphology. For this project we aim to divide the sources into the most typical and most exotic morphologies with the goal of semi-automated location of those with exotic morphologies.

Apart from the intrinsic morphological differences between observed AGN, differences may also often be partly due to the observation angle (Bianchi, Maiolino, and Risaliti 2012). Viewing the accretion disk from edge-on the jets will be most clearly visible, while the shorter wavelength emission of the core may be obscured by the torus. Face-on the core and accretion disk will be more visible but the jets will be more distorted and the morphology more difficult to determine due to the geometry and Doppler effects. At these angles the fact that one jet is at relativistic velocity toward the telescope while the other away from it may result in Doppler boosting and the near jet blueshifted due to the relativistic velocities while the far-side jet is redshifted. For particles moving at relativistic velocities the emitted wavelengths appear shorter or longer as the particle moves towards or away from a wave emitted in the direction of the observer before emitting the next, adding its velocity relative

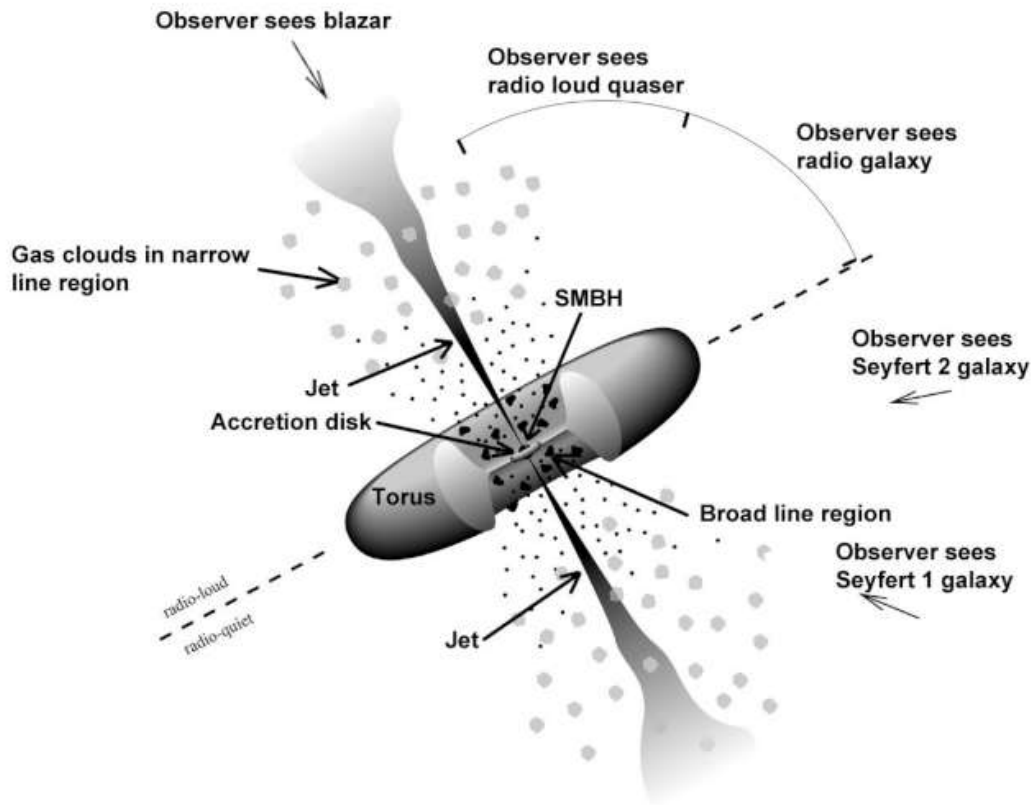


FIGURE 2.2: Unified AGN model showing the supermassive black hole, accretion disk, torus and jets as well as showing how the viewing angle changes the way the source may be observed. A quasar, or quasi-stellar object, are extremely luminous AGN. Seyfert galaxies are similar to quasars but with clearly detectable host galaxies. Blazars appear brighter due to their orientation with the jet directed nearly towards the observer. Image credit: Fermi and NASA.

to the telescope, per period of the emitted frequency, to the wavelength. The relativistic velocities of these particles may result in relativistic beaming at high angles to the disk, where two jets of near equal intrinsic brightness have observed luminosities that vary greatly due to their different viewing angles. These factors mean that the viewing angle is very important to the appearance of the observed galaxy, as is shown in Figure 2.2.

Radio galaxies are typically classified into a number of categories based on their morphology with the most typical being into two classifications, proposed by Fanaroff and Riley (1974). Sources that are brightest at their core, with brightness decreasing along the length of the jets, are classified as type FRI. Those with large, bright lobes appearing instead towards the ends of the jets where they collide with the intergalactic medium (the low density, hot plasma between galaxies) are classified as FR II. To do this the ratio of the distance between the brightest point in each jet to the overall length of the source is taken. If it is below 0.5 the radio galaxy is classified as FRI. If it is above 0.5 it is classified as FR II (Fanaroff and Riley 1974). Examples of each are reproduced in Figure 2.3 where the difference between the FRI and FR II type jets may be clearly seen. Figure 2.4 shows examples from the data used in this project.

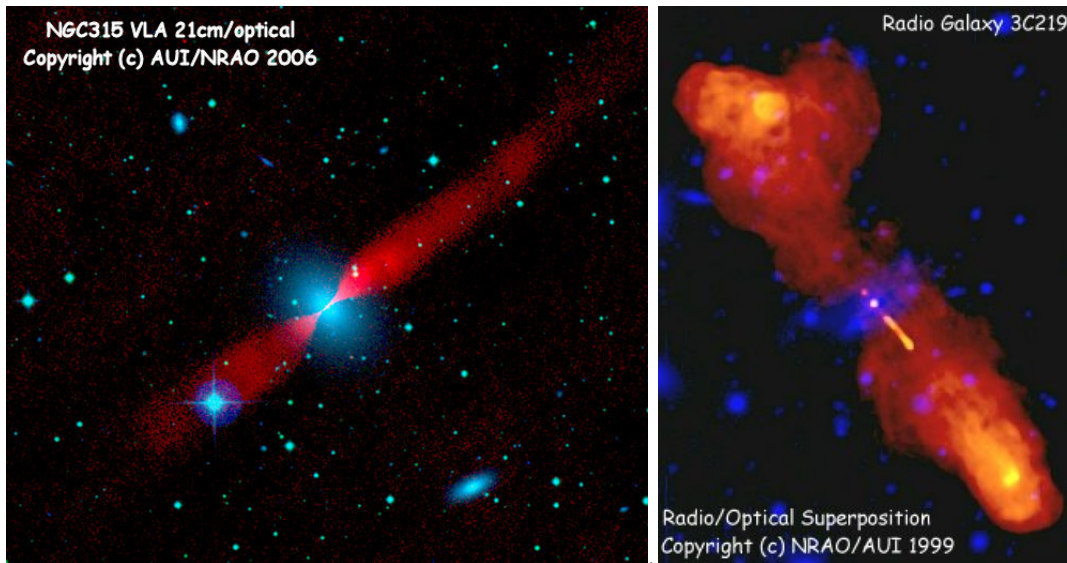


FIGURE 2.3: (Left) A typical FR I type radio galaxy. (Right) A typical FR II type radio galaxy. Image credit: AUI/NRAO, Bridle (2006)

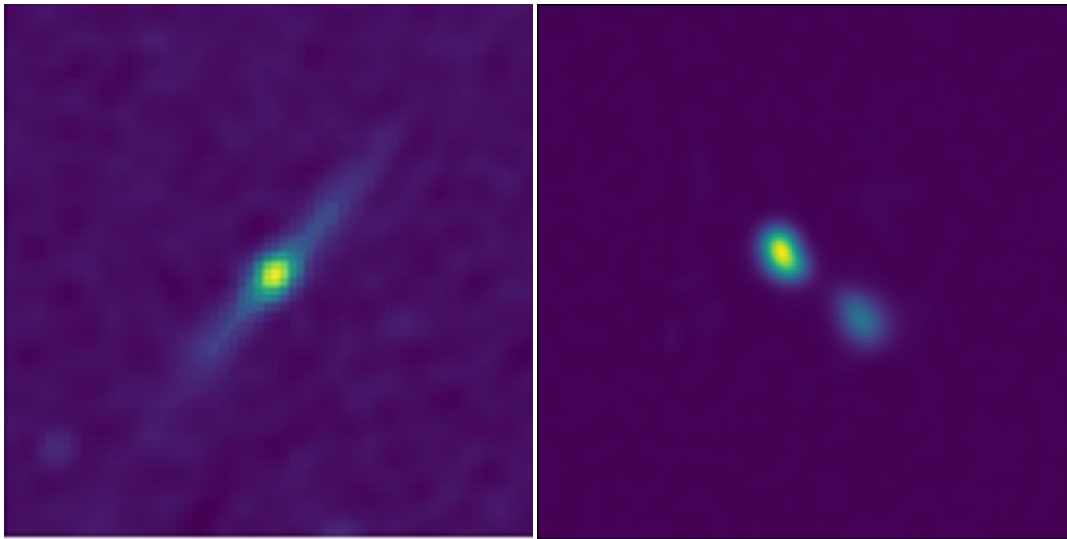


FIGURE 2.4: (Left) A typical Fanaroff and Riley type I radio galaxy where the brightness of the jets clearly decreasing along their length away from the core. (Right) Fanaroff and Riley type II radio galaxy showing the two large, bright lobes where the jets collide with the intergalactic medium. (Images found in MGCLS)

Although a morphological classification, intrinsic differences between the classes have been found. Fanaroff and Riley (1974), noted a sharp division in luminosity between the two classes with FRIIs having a higher luminosity than FRIs. Figure 2.5, reproduced from Owen and Ledlow (1994), shows how the relationship between the absolute isophotal magnitude (integral of light within a given brightness contour) and the radio luminosity differs for FRIs and FRIIs. Zirbel and Baum (1994), analysed the relationship between emission line luminosity and radio power and found correlations for FRI and FRII type galaxies independently but with the correlations offset with FRII galaxies showing about ten times the emission line activity.

Radio galaxies also interact with their environment. Morganti et al. (1988), looked at the effects of gaseous environments and, for radio lobes, found an inverse relationship between the radio structure size and cluster gas central densities. Croston et al. (2005), compared later X-ray Multi-Mirror Mission (XMM-Newton) observations of the X-Ray emitting hot gas of several FRI and FRII radio galaxies. For FRI galaxies they found that the distribution of the hot gas determines the radio-lobe morphology and evidence that subsonic expansion of the lobes heats the surrounding gas. For FRII galaxies the lobes are in equipartition and in pressure-balance with the surrounding gas. Mingo et al. (2022) analysed the relationship between the morphology and the accretion mode of the AGN, which can be either Radiatively Efficient (RE) or Radiatively Inefficient (RI). Although two thirds of FRIIs were RI they found that the relationship between the accretion mode and morphology is very indirect, the host galaxy environment controlling both in different ways.

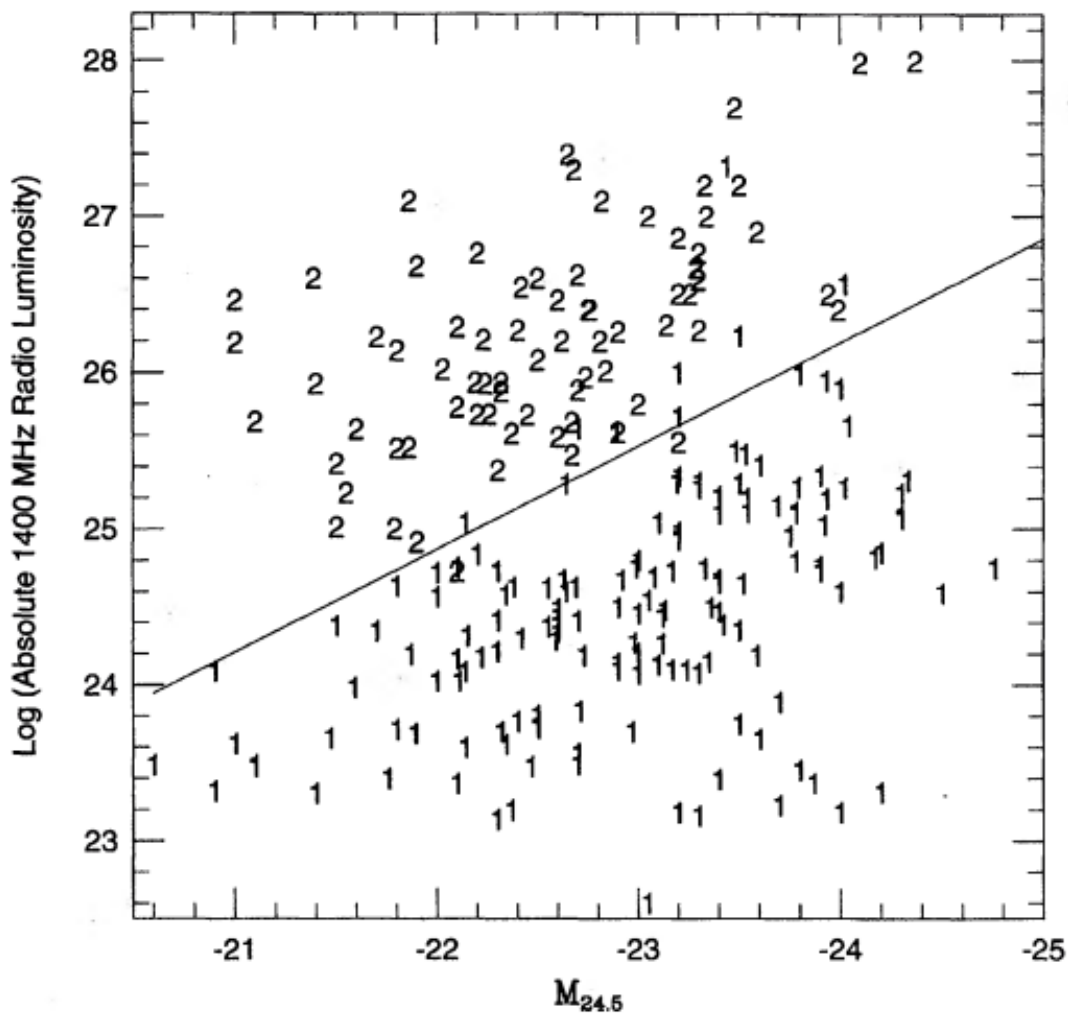


FIGURE 2.5: Reproduced from Owen and Ledlow (1994). The luminosity at 1400 MHz is compared to the optical absolute magnitude for a selection of FRI ("1" markers) and FRII ("2" markers) galaxies. The R-Band isophotal magnitude system, as described in Owen and Laing (1989), is used.

Apart from the typical FRI and FRII sources found above a certain flux-density luminosity, as discussed above, unusual morphologies of radio galaxies can be found.



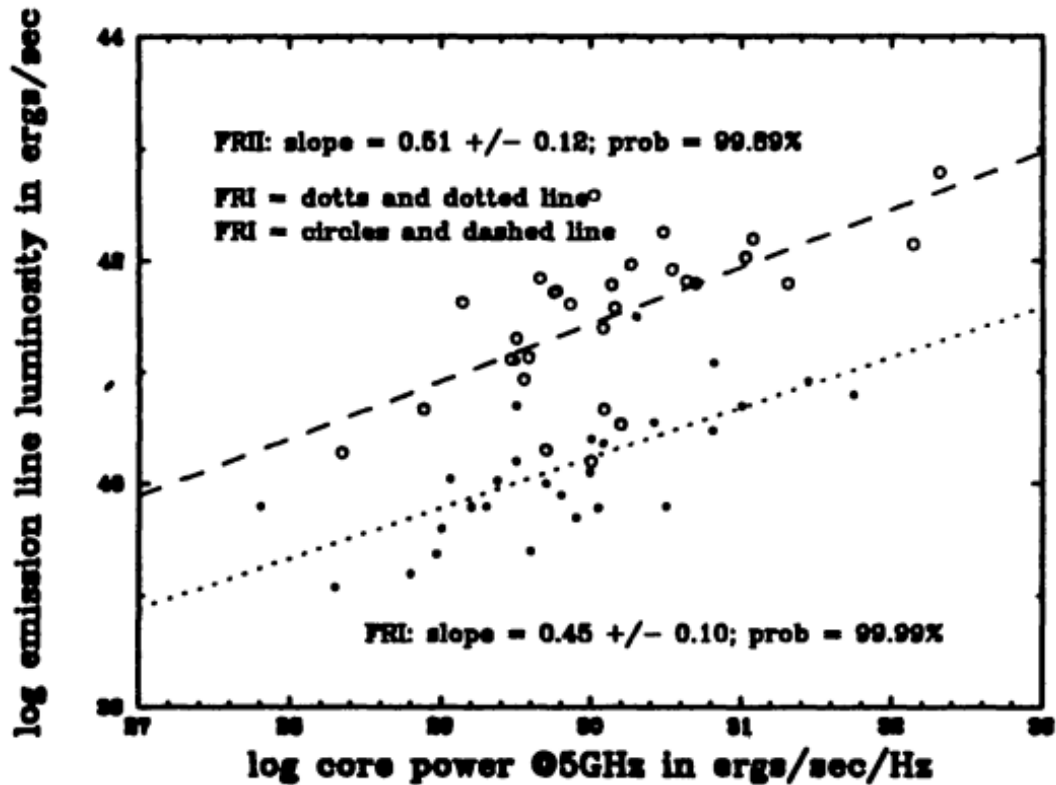


FIGURE 2.6: Reproduced from Zirbel and Baum (1994). The radio core power is correlated with the emission line luminosities. Power is measured as the power directly arising from the radio core, as distinct from the total power which includes that arising from energy deposited into the lobes. Erg is a unit of energy equal to  $10^{-7}$  J. It can be seen that a strong correlation exists for FRI and FRII type radio galaxies independently with the FRII radio galaxies consistently producing about ten times the emission line luminosity for a given core power.

These unusual morphologies may result from interaction with the Inter-Stellar Medium (ISM) or Inter-Galactic Medium (IGM). They may also arise due to episodic accretion, previous mergers with other galaxies, the precession from the orbit of binary black holes or spin-axis changes resulting from their merger which occurs when two massive galaxies merge as every massive galaxy appears to host at least one black hole at its centre (Kormendy and Richstone 1995). The merger of two supermassive black holes may result in helical or spiral shaped sources as they precess about each other (Begelman, Blandford, and Rees 1980). This can similarly result in Z- or S-shaped sources as the jets change orientation due to the precessing black hole. X-shaped sources, being radio galaxies that have what appear to be two pairs of offset lobes, have been observed and these may possibly arise in a number of different ways. These include merging AGN, when such a merger causes a drastic change in the spin of AGN, called a spin-flip, as well as hydrodynamical interaction of the jets with their surrounding medium (Leahy and Williams 1984; Rottmann 2001; Gopal-Krishna et al. 2012). Leahy and Williams suggest three models in which interaction with the medium may distort the source, shown in Figure 2.7, reproduced from Leahy and Williams (1984). In (a) an imbalance in the gas leads to a backflow. In (b), motion through the IGM creates a ram-pressure gradient that creates a bent source (ram pressure being the pressure exerted on an object moving in a fluid due to the

relative bulk motion of the fluid). Finally, in (c), if the source previously had jets along an axis differing substantially from the new axis then the old cocoon provides a channel for backflow where the jets are at lower pressure than the surrounding gas. An example of an X-shaped source created by hydrodynamical backflow can be seen in Figure 2.8, reproduced from Cotton et al. (2020), in which a recent 1.4 GHz MeerKAT observation of PKS 2014-55 is presented. Here PKS refers to the position-based naming of the source in the Parkes Southern Radio Source Catalog (Bolton, Savage, and Wright 1979; Wright and Otrupcek 1990). The diffuse radio emission in merging clusters creating halos and mini-halos around powerful radio galaxies in cooling core clusters as well as radio relics from cluster emission and old radio lobes is also observed (Ferrari et al. 2008). Figure 2.9 shows three of the more unusual sources found in the MGCLS data.

## 2.3 The MeerKAT Galaxy Cluster Legacy Survey

The survey data used in this project was observed using the MeerKAT radio telescope. MeerKAT is one of the most sensitive radio interferometer of its class. Due to its high sensitivity of  $\sim 3\text{-}5 \mu\text{Jy}/\text{beam}^{-1}$  the survey has many detailed and diffuse images of interesting radio sources that we want to find. However, this also means that the fields are crowded with many point-like sources and that many sources may have some complex structure that the machine learning algorithms may struggle with.

The MeerKAT radio telescope, an extension of the Karoo Array Telescope (KAT), consists of 64 13.5 metre offset-Gregorian dishes, with a minimum and maximum baseline of 26 m and 8 km respectively, making it sensitive to large-scale, low surface brightness features at a few arcmin scales, while still achieving few arcsec maximum angular resolution. A compact ( $\sim 1$  km) inner core containing 70% of the dishes is surrounded by the remaining dishes that are spread out over the much wider distances up to the maximum baseline of 8 km. The layout of the dishes is shown in 2.10. It can observe in the UHF (580–1015 MHz), L (900–1670 MHz) and S (1.75–3.5 GHz) radio bands.

Eventually, MeerKAT will form the compact core of the SKA in South Africa, SKA-Mid, which will extend to a maximum baseline of 150 km. The first phase, SKA1-Mid, will have 197 15 m antennas that will observe over five frequency bands in the 0.35–15.4 GHz range (Swart, Dewdney, and Cremonini 2022).

The data for this project comes from the MeerKAT Galaxy Cluster Legacy Survey (MGCLS, Knowles et al. 2022) which used the MeerKAT telescope to survey 115 galaxy clusters in high fidelity long-track observations in the L-band (900-1670 MHz). Each cluster field was observed for about 6 to 10 hours in full polarisation mode. The complete catalogue presented in Knowles et al. has over 600 000 sources, representing a challenging task for manual classification. A central observing frequency of 1.28 GHz with typical sensitivity ranging from about 3 to 5  $\mu\text{Jy}/\text{beam}$ . The images are sensitive to structures up to about 10 arcmin scales. The basic full-field spectral cubes span about  $2 \text{ deg} \times 2 \text{ deg}$  while the enhanced products consist of the inner  $1.2 \text{ deg}$  by  $1.2 \text{ deg}$  field of view corrected for the primary beam. A wide field of view means there are hundreds to thousands of non-cluster sources

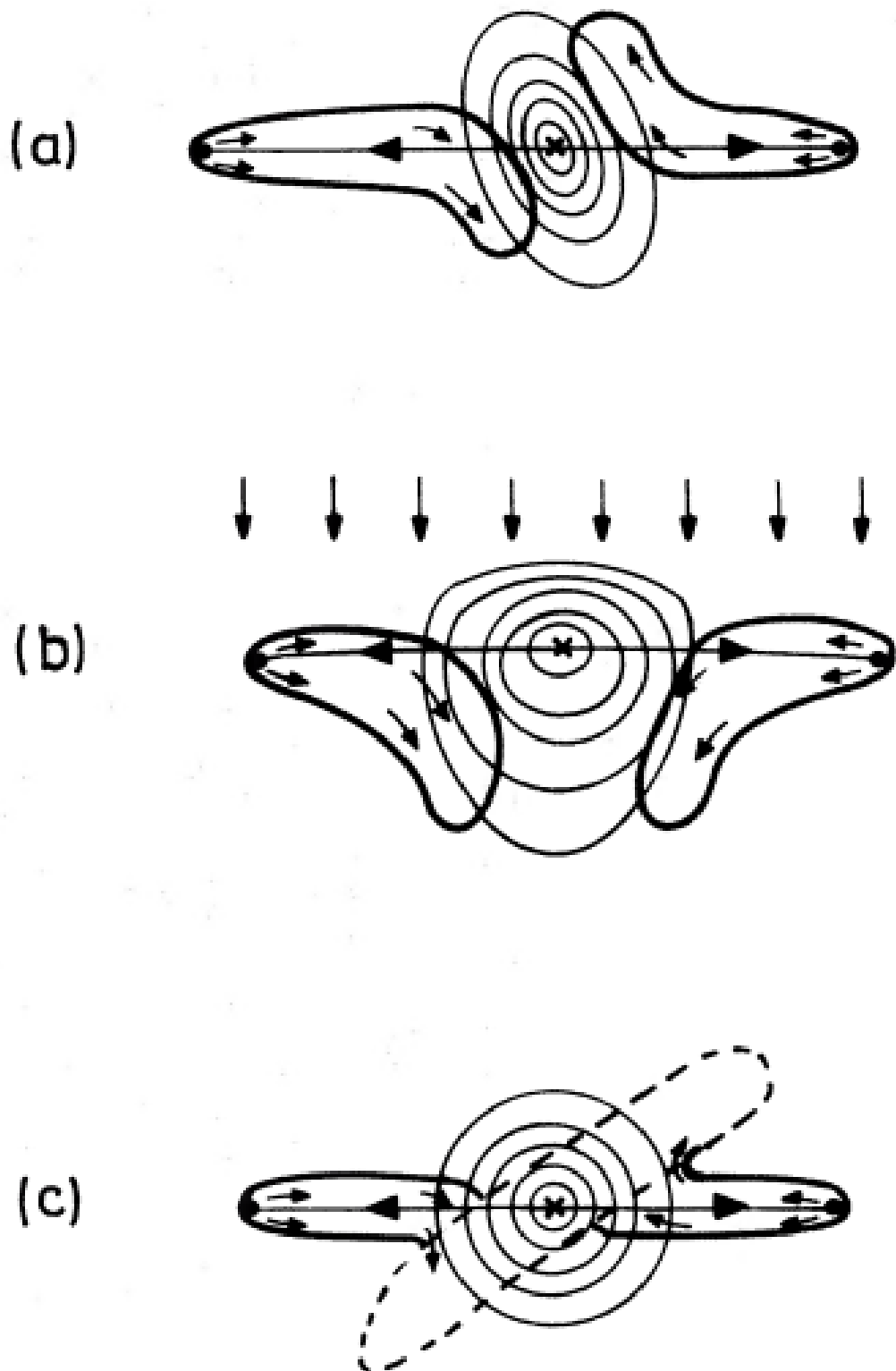


FIGURE 2.7: Reproduced from Leahy and Williams (1984). Diagrams of three distortion mechanisms. The thin lines represent x-ray contours that indicate the surrounding gas. The thick lines represent the radio jets. The arrows indicate flow. The "x" indicates the AGN. The dashed outline in (c) represents the cocoon of the old jets.

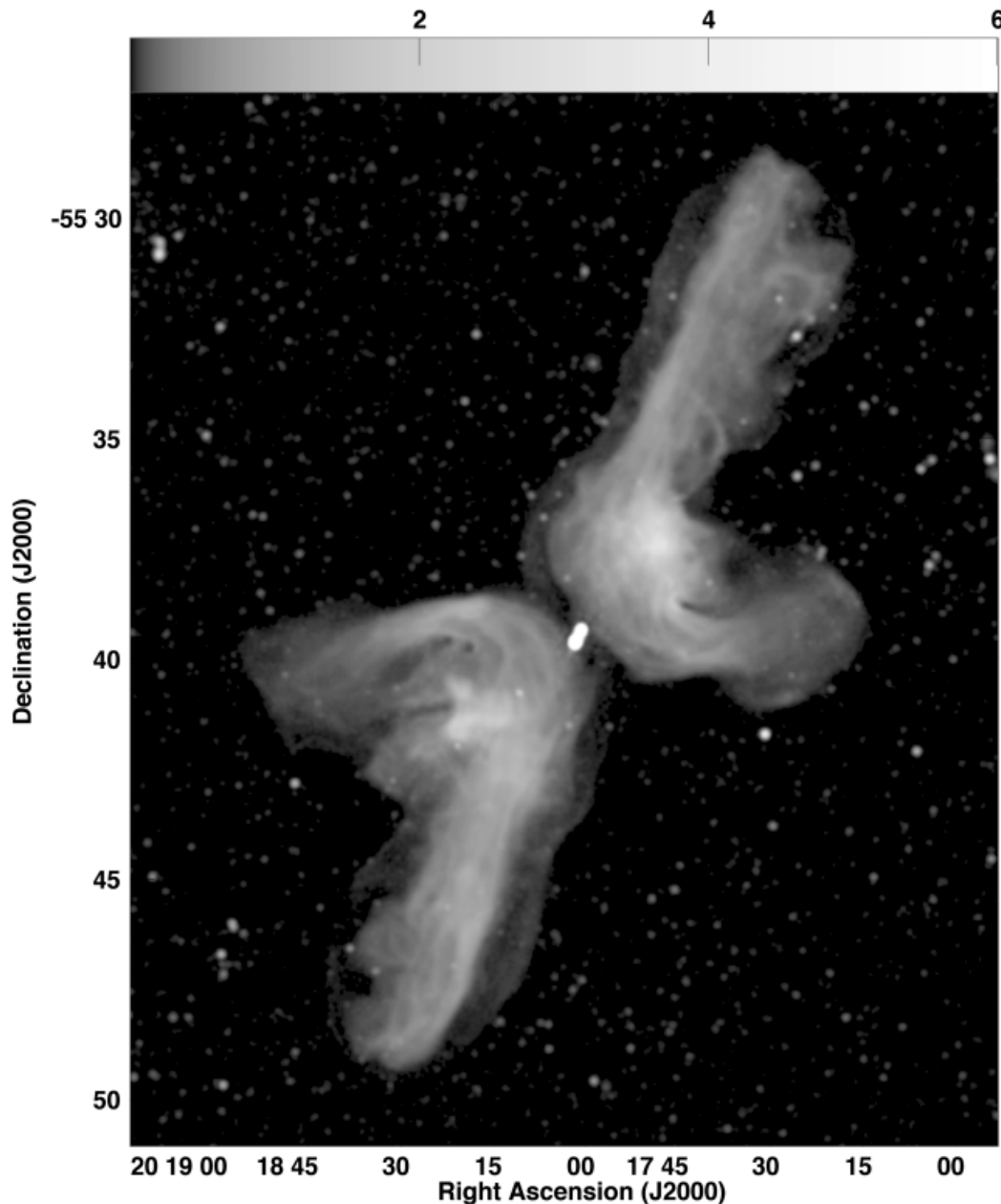


FIGURE 2.8: Reproduced from Cotton et al. (2020). A MeerKAT observation of PKS 2014-55 showing an example of an X-shaped radio galaxy created by hydrodynamical backflow.

per image. All of the images used in this project are from this survey. However, the approach could naturally be applied to other MeerKAT images.

Figure 2.11, which is reproduced from Figure 1 from Knowles et al. (2022), shows the sensitivity of the MeerKAT telescope as it was used for this survey. It also shows some of the other capabilities of the telescope and the data which are not used for this particular project, but demonstrate the power of the telescope. Panel A shows the sensitivity of MeerKAT by showing three cutouts, first the full resolution image is dominated by compact sources. It is convolved to 25" to increase sensitivity to larger scale structure, as shown in the second image. However the diffuse large scale structure is hidden by the blended compact sources. To solve this problem the

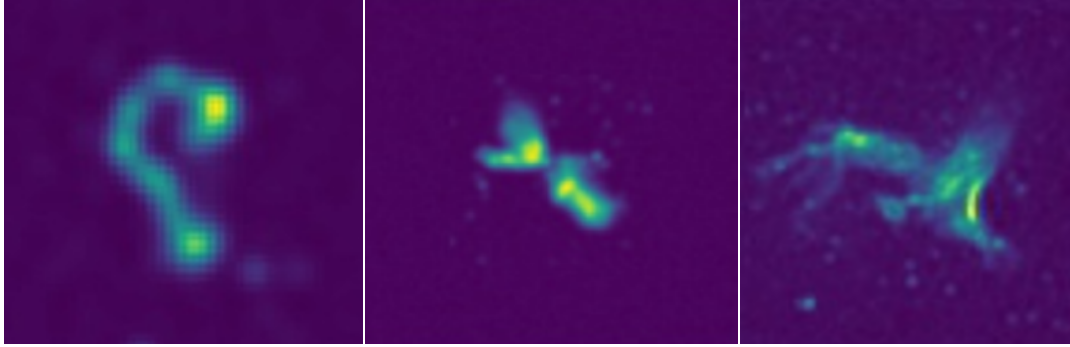


FIGURE 2.9: (Left) A bent tail radio galaxy. (Centre) An X-shaped radio galaxy. (Right) Galaxy cluster emission also comes up relatively often in this data. They have been labelled exotic for the purposes of this project. (Images found in MGCLS)

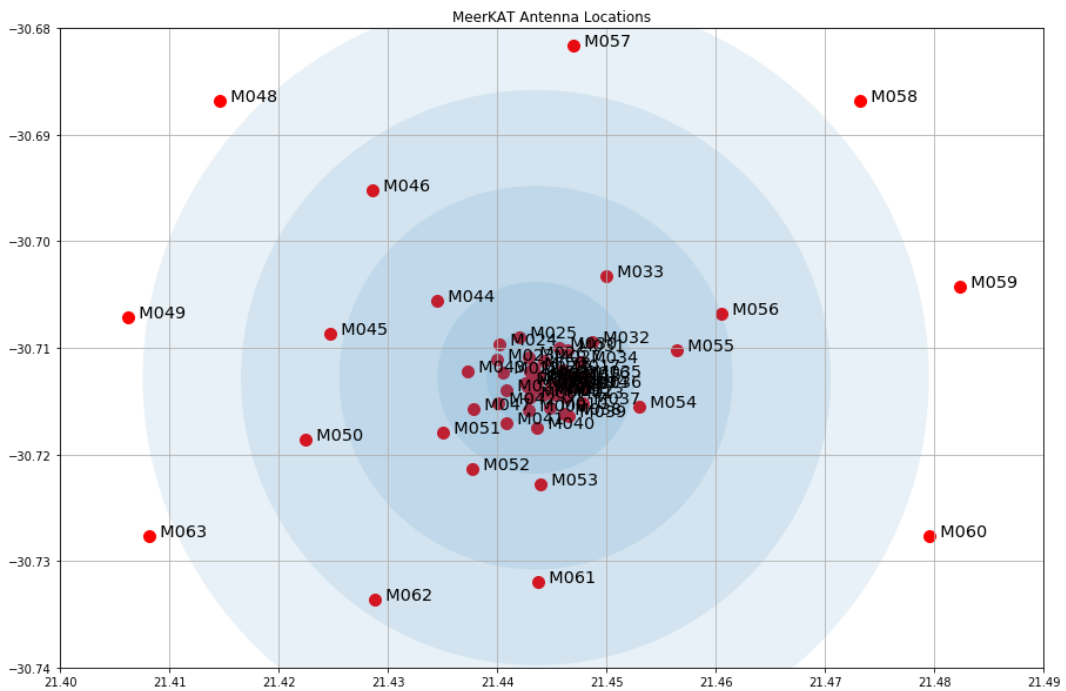


FIGURE 2.10: Reproduced from Goedhart, Krishnan, and Camilo (2022). The layout of the 64 MeerKAT antennas. Coloured rings are about 1, 2, 4, 6 and 8 km in diameter.

authors filtered out the small scale structure using the technique described in Rudnick (2002), a simple multiresolution filtering technique, which produced the third image in which the larger scale structure is clearly visible. Panel B shows an example of how the wide 0.8 GHz bandwidth at L-band of MeerKAT allows for in-band spectral analysis. Panel C shows an example of how a Rotation Measure (RM) (a polarisation rotation due to a magnetic field) may be determined due to the full polarisation and sensitivity of the observations. And Panel D shows an example of resolved HI detection.

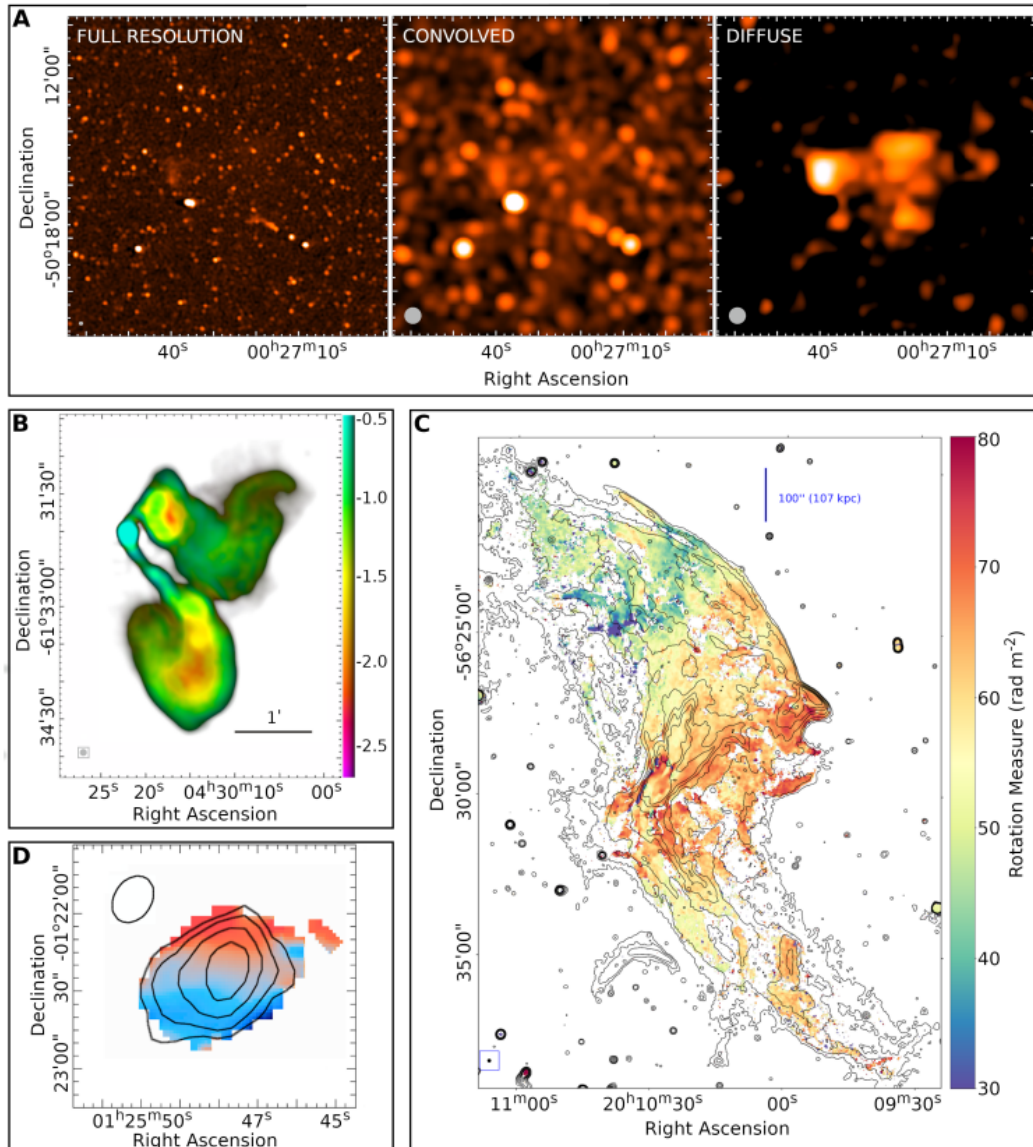


FIGURE 2.11: Reproduced from Knowles et al. (2022), showing capabilities of the MGCLS data. (Panel A) Brightness cutouts from field MCXC J0027.3-5015 showing instantaneous sensitivity of MeerKAT at a range of scales. Left to right, the full resolution image at  $7.4 \times 7.0$  arcsec, then convolved at 25 arcsec, finally the filtered diffuse emission at 25 arcsec resolution. (Panel B) Example of an in-band spectral map of a bent tail source from field MCXC J0431.4-6126. (Panel C) Rotation Measure (RM) map of a complex source from field Abell 3667. (Panel D) An H1 velocity map of Minkowski's object in Abell 194 at a resolution of  $19 \times 15$  arcsec.

## 2.4 Neural Networks

Neural Networks (NN, McCulloch and Walter 1943; Rosenblatt 1961) form the basis of the machine learning approaches used in this project. NNs consist of artificial neurons which can take some input, such as an array of values, and produce some output value based on their design. These neurons use two types of parameters to

determine the output. First, the various inputs are considered to have varying significance, called their weight. Second, the neuron itself has a bias value that will bias it towards certain outputs. These neurons form a network of multiple layers. The first layer, the input layer, does not consist of neurons in the typical sense and simply contains the input data. The number of neurons in this layer is determined by the dimensions of the input for which the network is being designed. Likewise, the last layer, the output layer, must contain the number of neurons of the dimensionality of the desired output. Between these there can be any number of layers of typical neurons, called hidden layers. A neuron in a hidden layer will take as input any or all of the outputs of the previous layer and will feed its own output as an input to the subsequent layer. An example is shown in Figure 2.12, reproduced from Nielsen (2015). This type of network is called a feedforward network as the input for any layer only depends on previous layers and not subsequent layers.

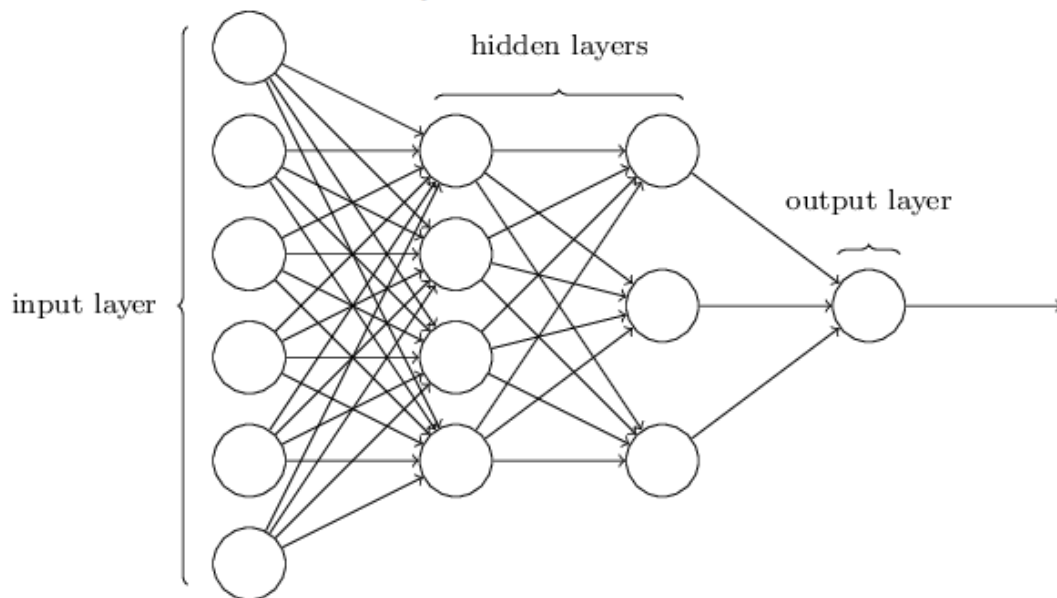


FIGURE 2.12: Reproduced from Nielsen (2015), a basic example of a Neural Network. Each circle represents a neuron. The input layer is on the far left, and takes an array of six input values and provides them to the first hidden layer. There are two hidden layers, of neurons of four and three neurons each. Finally, the output layer takes three values from the last hidden layer as input and provides a single value as output.

The function that determines the output of the neuron from the input and parameters is called the activation function. The first neural networks, called Perceptrons (Rosenblatt 1957), used a simple activation function that would produce either a 1 or a 0 (whereby the neuron could be said to have "fired" or "not fired") by comparing the weighted sum of the inputs to some threshold value. More modern neural networks, however, use other activation functions, such as sigmoid neurons, that can produce any value between zero and one and move smoothly between these outputs. These rely on the sigmoid function, defined by

$$\sigma(z) \equiv \frac{1}{1 + e^{-z}}, \quad (2.1)$$

for some value  $z$ , typically taken as the inputs multiplied by their weights and added to the bias, where  $e$  is the natural exponent.

This allows for small changes in the bias and weights of a neuron to produce small changes in the output for a given input rather than a sudden flip between 0 and 1. This is desirable when training the network, the process of gradually adjusting weights and biases of the network so that it produces the desired outcomes, as it allows small adjustments to make smaller changes to the output of the network without drastically and unpredictably changing the behaviour.

## 2.5 Convolutional Neural Networks

The machine learning approach used in this project to search through the data are convolutional autoencoders. To understand how these work we must first look at a type of neural networks called Convolutional Neural Networks (CNN, LeCun et al. 1989). CNNs will form the basis of the two parts of the eventual autoencoders. CNNs are used as they are particularly good at working with images, or other data that is arranged into multidimensional grids of values rather than arrays, as the goal is to search for exotic sources by morphology. The autoencoders consist of two multi-layered CNNs, the encoder and the decoder, which must encode images into an encoded form and attempt to restore the original image from that encoded form respectively.

Convolution is an operation on two functions. This can consist of some function on interest,  $f$ , and some manner of weighting function that, at a specific point, allows us to calculate a weighted average of the  $f$  about that point. Equation 2.2 defines a convolution,  $S$ , of such a functions,  $f$ , with a weighting function,  $w$ , at a point  $x$ ,

$$S(x) = \int f(a)w(x - a) da \quad (2.2)$$

where  $a$  is some point such that  $w(x - a)$  provides a weighting based on the distance from  $x$ . The convolution of these functions is often denoted  $(f * w)$ . For discrete arrays, such as pixels in an image, this becomes a summation.

$$(f * w)(x) = \sum_{a=-\infty}^{\infty} f(a)w(x - a) \quad (2.3)$$

For a multidimensional array, such as a two dimensional array of  $m$  pixels, it becomes a summation over both dimensions. If the value of the functions is taken as zero everywhere outside the image then it may be calculated by summing over the dimensions of the image. Here the two dimensional image is often called the Input,  $I$ , the second function is called the kernel,  $K$ , and the resulting convolution over a two dimensional discrete array of pixels is

$$(I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (2.4)$$

where  $i$  and  $j$  are the indices of the pixel being convolved. The kernel is usually represented as a multidimensional array of parameters that is adjusted during the training process. The output of this is often called a feature map. The kernel does not need to be the same size as the input and indeed the use of small kernels may provide good computational performance and is the most common scenario in practice.



The use of convolution also means that the same weights are applied throughout the image, providing some translational equivariance.

The kernel should be selected so that its convolution with an input image produces some useful result. For instance it may be chosen to weight a certain pattern of pixels more highly than others, which could potentially be useful in detecting that pattern. However it is difficult to select such a kernel and so training is used to adjust the elements of the kernel, called weights, until they approximate the desired result. Numerous input images for which the desired result is known are used to adjust the weights by comparing their actual results with the desired result and making appropriate adjustments to the weights until the desired and actual results are sufficiently similar.

Training of the weights can be done using an algorithm such as backpropagation (Werbos 1974; Rumelhart, Hinton, and Williams 1986), which is described in more detail in Section 2.6. Multiple convolutional layers may be used in constructing the CNN by using the result of the convolution with one kernel as the input for the next. Additionally, different types of layers may be combined with convolutional layers that perform different functions. For example, pooling layers, described in Section 2.7, perform a type of dimensionality reduction.

## 2.6 Training

Training the weights of the kernels is done using a gradient descent based approach. Given some function,  $f$ , we wish to minimise, calculating the derivative or gradient indicates how to alter its parameters in order to further reduce it. The function in this case is some cost or objective function that measures performance of the model. Its evaluation relies on multiple weights so it is a multivariate function that must be minimised. Therefore the derivative may also be a multivariate value that must be minimised in order to determine the best direction to move in. Once the proper direction is found, being the direction along which the directional derivative is minimised, the values may be adjusted in that direction in an appropriately determined increment. The size of the change is modulated by a hyperparameter called the learning rate. The method is not perfect as it does not properly account for local minima and saddle points but it generally can return a sufficiently good result, in the context of deep neural networks, for cost functions that are not complex or multi-modal.

As most training sets are too large to evaluate the gradient with all samples, Stochastic Gradient Descent (SGD) breaks the training samples into uniformly drawn batches (more formally this is referred to as mini-batched SGD, but we will use SGD to mean mini-batching in this project). By simply using one batch to evaluate the derivative a sufficiently good estimate may be made in order to facilitate effective optimisation progress. Since the batch size can be held constant with a growing number of training samples, the computational time required to take one step may be constant for any training set. Once all of the samples in the training set have been used in batches it is said that the weights have been trained for one epoch. Typically, depending on the size of the training set, the algorithm is trained for numerous epochs allowing all of the training samples to influence the weights multiple times (Goodfellow, Bengio, and Courville 2016).

In order to determine the gradient for one training step, a method called Back propagation (Werbos 1974; Rumelhart, Hinton, and Williams 1986) is used. Before the back propagation step can be taken the cost for some sample must be found using forward propagation. Forward propagation is when some sample is passed through the neural network and some cost,  $C$ , is derived from the difference between the output and the intended output. In the case of autoencoders the intended output is the input image so no supervision is required to provide the correct answer and the log error is simply taken.

For some arbitrary function,  $f$ , the back propagation step will calculate  $\nabla_{\vec{x}} f(\vec{x}, \vec{y})$ , where  $\vec{x}$  is a collection of variables whose derivatives we need and  $\vec{y}$  is a collection of additional input for which the derivatives is not required, such as the target vector. In this case this will be some manner of cost function with the parameters that need to be adjusted. In a deep network, the various neurons with their biases can be treated as the edges on a graph and directed edges, having the weights, added where one depends on another. This graph then represents the neurons of the network and the connections between the layers. Let the biases at each neuron be  $b_j^l$  with  $l$  indicating the layer and  $j$  the specific neuron so that the biases of layer  $l$  may be indicated with the vector  $\vec{b}^l$ . Similarly, let the vector  $\vec{a}^l$  indicate the activations of layer  $l$  for some arbitrary input. The activation of a neuron is its output for some input and depends on its inputs and bias. Finally, let the weights of the previous layer,  $l - 1$ , be denoted with the matrix  $w^l$  so that  $w_{jk}^l$  indicates the weight of the of the neuron  $k$  from the layer  $l - 1$  feeding into the neuron  $j$  in layer  $l$ . The activation for a given layer,  $l$ , for some function,  $f$ , will be

$$a_j^l = f\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right), \quad (2.5)$$

over all the neurons,  $k$ , in the layer  $l - 1$ . Written as vectors, this becomes

$$a^l = f(w^l a^{l-1} + b^l). \quad (2.6)$$

Let  $w^l a^{l-1} + b^l$  be denoted as  $z^l$  for convenience. Each neuron,  $j$ , in each layer,  $l$ , is also associated with some error,  $\delta_j^l$  with  $\delta^l$  being the error of the layer  $l$ . In order to correct this error we must adjust the weights and biases. The adjustment that must be made to each individual weight or bias is given by the partial derivative of the cost in terms of that weight or bias,  $\frac{\partial C}{\partial w}$  or  $\frac{\partial C}{\partial b}$ . We can say that  $\frac{\partial C}{\partial z_j^l}$  is a measure of the error in a neuron as it is what is actually passed into the function. We can then define the error of that neuron as

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l}, \quad (2.7)$$

for neuron  $j$  in layer  $l$ . Let the output layer be denoted  $L$ . The error for the output layer is then given as

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} f'(z_j^L), \quad (2.8)$$

which is simple to calculate as the activations of the output layer is the output of the network (Nielsen 2015). This may be written in matrix form as

$$\vec{\delta}^L = \nabla_{\vec{a}} C \odot f'(\vec{z}^L), \quad (2.9)$$

where the operator  $\odot$  represents the Hadamard product, an element-wise product of two vectors that produces a vector of the same length. Next, backpropagation requires finding the error of a layer,  $l$ , in terms of the following layer,  $l + 1$ . This is given by

$$\vec{\delta}^l = ((w^{l+1})^T \delta^{l+1}) \odot f'(\vec{z}^l), \quad (2.10)$$

where  $(w^{l+1})^T$  indicates the transpose of the weight matrix. Equations 2.9 and 2.10 together allow for the error of any layer to be calculated. First the error is determined for the output layer using 2.9, after which every previous layer may be calculated in turn using 2.10. At this point it must be determined how much of the error is attributable to the weights and how much to the biases, so that they may be adjusted. For the biases we have that

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l, \quad (2.11)$$

and for the weights we have that

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l. \quad (2.12)$$

So first the feedforward step in which all the values of  $z$  and  $a$  are determined. Then the output error may be determined as shown in Equation 2.9. The error is then backpropagated through the layers as shown in Equation 2.10. From here the gradient of the cost function may be determined as shown in Equations 2.11 and 2.12. At this point the weights and biases may be updated with gradient descent according to the learning rate and other hyperparameters.

First the feedforward step, finding the cost and also building the graph of neurons. A table is used to store the derivatives of nodes that have already been calculated. It will then work its way back along the graph, in the opposite direction of the forward propagation, calculating the partial derivative at each node using chain rule and storing it in the table. If multiple paths exist back to the same node they are merely summed at that node. Many variations exist for specific implementations or to improve efficiency, but now that the derivatives have been calculated the weights can be updated as aggressively as specified by the learning rate. (Goodfellow, Bengio, and Courville 2016)

## 2.7 Pooling Layers

Pooling layers reduce the dimensions of the data and replace a subset of the image with a single value, such as reducing a square patch of pixels down to a single pixel. Pooling layers can be used to reduce the dimensions of the input data and so constrain the size of the encoded form. This can force the autoencoders to try and learn some more non-trivial features of the sources. Ideally the features from typical sources will not always be found in the exotic sources. This would mean that the

exotic sources might be more poorly reconstructed allowing them to be identified.

A common type of pooling is max pooling (Zhou and Chellappa 1988). In the max pooling technique the maximum value is taken from any input elements within a certain rectangle that must be reduced down to a single element. To reduce the size of an image by a certain factor the pixels are divided into rectangular patches of that size and an output produced of the maximum value of each, as is shown in Figure 2.13. More information about other types of pooling can be found in Boureau, Ponce, and Lecun (2010), Lee, Gallagher, and Tu (2016), and Pala et al. (2018).

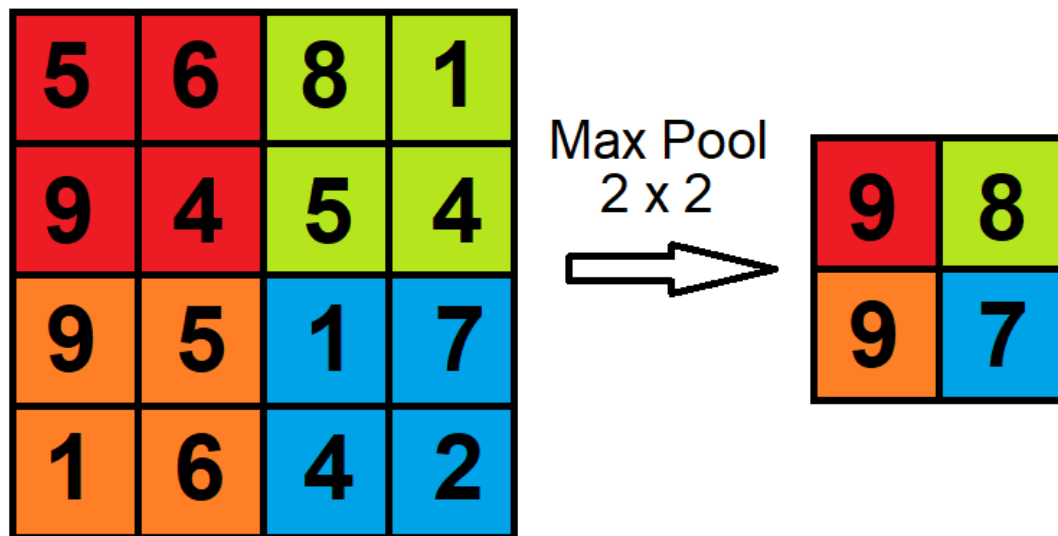


FIGURE 2.13: An example of a max pooling operation that reduces a  $4 \times 4$  input by a  $2 \times 2$  filter. The maximum value is selected in each region.

Spatial Pyramid Pooling (SPP) in convolutional networks (He et al. 2014) is a different pooling technique that attempts to preserve spatial data within the image while pooling images of arbitrary size or resolution. The image is pooled down to a fixed size that is suitable for certain types of layers requiring fixed input sizes. SPPs make use of spatial matching, an improvement of pyramid matching described below, to pool inputs of varying sizes and resolutions while maintaining spatial information.

Pyramid matching (Grauman and Darrell 2005) finds an approximate correspondence between two sets of vectors in feature space (the space containing the input whether this is the original image or extracted features). This is done through a series of grids imposed on the space at varying resolutions. For each cell in each grid the number of matching features is the number of vectors that appear in that cell for both sets. The number of matching features is summed over for all cells in the grid to find the total number of intersecting points in that grid. This is done for all grids at the varying resolutions. The points that match in the finer grids also naturally match in all coarser grids. So only the new matches at each increasingly coarse grid level are considered. The number of matches for each grid is then weighted according to the coarseness of the grid, with the finest grid having the highest weighting and the coarsest grid having the lowest. The sum of these weighted totals provides a measure of similarity between the images, although that is not needed here as merely

the technique for partitioning images is used.

A spatial matching approach (Lazebnik, Schmid, and Ponce 2006) improves pyramid matching in that it allows for spatial data to be used for matching but only for sufficiently similar features. It attempts to consider the spatial data by matching the features in image space. It also assumes that the features should only match with the most similar features. The feature space is quantised into  $M$  discrete types. For each of these sets image space coordinates of the features within that type are taken and pyramid matching performed. The effect on an image is to apply the grids at varying resolutions as described above to the image and aggregate the features in each subregion of each grid into  $M$  bins.

SPPs make use of the spatial matching approach to pool their input into spatial bins. The number of bins is constant for any input image as they have sizes proportional to the image size. As the bins are arranged spatially and the size of the bins is adjusted according to the input size this allows for the images of arbitrary size or resolution to have their features reduced down in a consistent manner while maintaining spatial information. Figure 2.14, reproduced from He et al. (2014), shows an example of such a layer and shows how the input is split into bins. In this example, there are three levels of pooling. The first layer has only a single bin, the second layer has four bins and the last layer has sixteen bins. In each spatial bin in each layer, the values of each filter, of which there are 256 in this example, is pooled using max pooling. So the single bin layer will pool each filter across the whole image, while the sixteen bin layer will pool these filters only within each of the sixteen spatial bins. As the number of bins and layers is fixed, the output size is also fixed regardless of the input size.

SPPs are useful in the case of classifying radio galaxies from such a survey as the various observed sources are of vastly differing sizes and distances which can lead to source cutouts being of varying sizes and resolutions. As these should not affect the classification of the source the use of pyramidal pooling layers may reduce the effect that this has on the encoded form.

## 2.8 Autoencoders

Now having discussed the layers that are used to build up autoencoders, the complete algorithms may be discussed. Autoencoders are a class of feedforward unsupervised machine learning algorithms which consist of a pair of neural networks which attempt to take an input image, encode it into a latent representation form using the first network and then decode that back into the original image with the second network, as show in Figure 2.15. These two models are called the encoder and the decoder, respectively. The particular networks used here are CNNs making these convolutional autoencoders. The difference between the input and reconstructed image may then be backpropagated through the network, allowing for unsupervised learning. Autoencoders were initially developed for feature learning and dimensionality reduction where the encoded latent form could be constricted to learn to encode certain features or constricted by size to learn a smaller representation of the input (LeCun 1987; Bourlard and Kamp 1988).

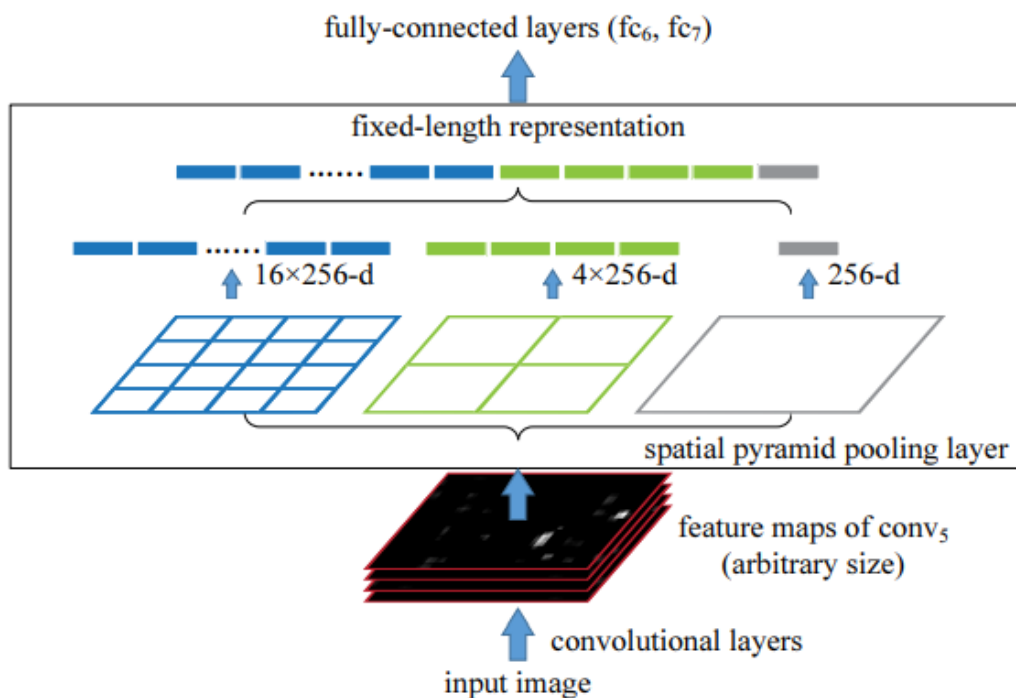


FIGURE 2.14: Reproduced from He et al. (2014). An example of a neural network with a spatial pyramid pooling layer between convolutional layers and fully connected layers. 256 is the filter number of the last convolutional layer in this example.

The encoder CNN must take the input images and try to map them to the latent space in some encoded form. The latent space being the space in which the encoded forms of input images, as output by the encoder, reside. Restrictions placed on this encoded form may induce certain desirable results. For example, a sparsity requirement may force the encoder to recognise common features in the data which may be useful for certain types of classifications. Restrictions on the size of the encoded form may cause the encoder to learn a form of compression or to isolate the most important features for reconstruction.

The decoder must then take this encoded form as input and try to restore the original image. The difference between the input and output images may be compared. Back-propagation of the error signal is used to update the weights of the autoencoders. In this way the algorithms are able to train in an unsupervised manner to map images to and from the latent space.

Training on more common morphologies may allow the rapid recognition of exotic morphologies in vast datasets. If test images are run through the trained autoencoder and the error of reconstruction is calculated, those most similar to the training data will have the lowest reconstruction errors as the encoders are well trained for them. But exotic sources will have higher reconstruction errors as the encoders have not been trained to reconstruct images with their potentially unique morphological structure. A classification for a given source may then be chosen by comparing the reconstruction error to some threshold.

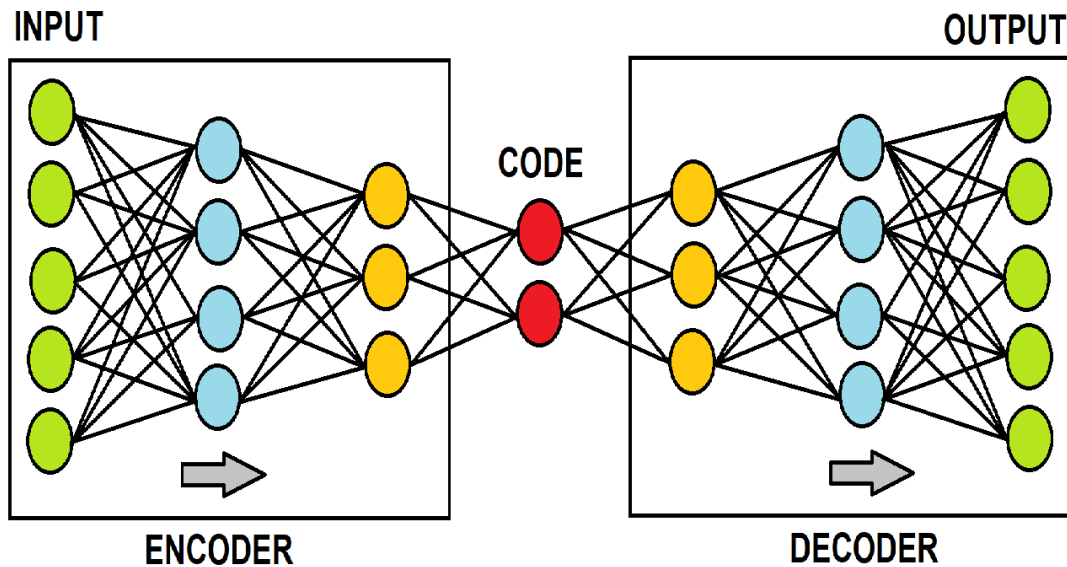


FIGURE 2.15: Basic structure of an autoencoder. The encoder reduces the input to a code form. The decoder attempts to reconstruct the original input from the code. Each layer is indicated with a different colour. The lines show how previous convolutional layers affect subsequent layers as an input image moves through the autoencoder, moving in the direction of the arrows. During backpropagation the error signal moves back along the lines against the arrows.

## 2.9 Performance Evaluation

Once the training is completed the performance of each model is measured in comparison. A number of metrics are used to do this.

When a model is intended to detect anomalies or specific instances of data it will return either a positive or negative prediction for each input to indicate if it has found what it is looking for. In this project the positive case will refer to anomalies while the negative case will refer to typical sources. These predictions could either be correct, called true, or incorrect, called false. So there are four potential outcomes for the classification of an input. If the input has been correctly identified it is either a True Positive (TP) or True Negative (TN). If it has been incorrectly classified it is either a False Positive (FP) or False Negative (FN).

The totals of these four values or their relative fractions or proportions may be arranged into what is called a contingency or confusion matrix. The layout is shown in Figure 2.16. The columns of the matrix are the actual labels while the rows are those assigned by the model. For the simple case of positive or negative, this means the table will have four cells. The cells running diagonally, then, from top left to bottom right are the cells in which the true and predicted values match and the prediction is correct. Other cells where the row and column do not match contain inputs incorrectly classified.

The recall parameter, or the True Positive Rate ( $TPR$ ),

$$Recall = TPR = \frac{TP}{TP + FN'} \quad (2.13)$$

		True Class	
		Positive	Negative
Predicted Class	Positive	True Positives	False Positives
	Negative	False Negatives	True Negatives

FIGURE 2.16: A confusion matrix for a binary classification.

is the proportion of positives that are correctly identified. In this case recall will be the proportion of positive inputs that are correctly identified by the model as positive for a given threshold. The recall is very important if we want to locate as many positive inputs as possible. Similarly, the False Positive Rate ( $FPR$ ),

$$FPR = \frac{FP}{TN + FP} \quad (2.14)$$

is the rate at which negatives are incorrectly identified as positives. Precision,

$$Precision = \frac{TP}{TP + FP} \quad (2.15)$$

is the proportion of those inputs marked positive which are truly positive. In classification or any application where the labels applied must be relied upon and be correct it is important. However, if the aim is to locate as many positive cases as possible a tradeoff of precision can be made for higher recall so that the user will locate more positives but may have to look through more candidates.

The harmonic mean of the precision and recall, called the  $F_{measure}$ ,

$$F_{measure} = F_1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (2.16)$$

is a way often used to measure the performance of a classifier. It assumes that the performance in terms of recall and precision are equally important. As we are trying to recover as many positives as possible, Recall is more important than Precision in our application area. For example, it would be preferred in our application for the



user to look through 10 sources of which 5 are exotic than to look through 5 sources of which 3 are exotic. A more generic  $F_{measure}$ , called  $F_{\beta}$ ,

$$F_{\beta} = (1 + \beta^2) \frac{Recall \times Precision}{(\beta^2 Precision) \times Recall'} \quad (2.17)$$

allows the relative weighting between Recall and Precision to be configured.  $\beta$  is a positive value that indicates the factor by which Recall is to be weighted against Precision. For example, if  $\beta$  is 0.5 that indicates that Recall is only half as important as Precision. If  $\beta$  is 2 then that indicates that Recall is twice as important as Precision. Lastly, if  $\beta$  is 1 then they are equally important, and this is then just the harmonic mean from Equation 2.16, (Tharwat 2020).

The Receiver Operating Characteristic (ROC) plots TPR against FPR (Hanley and McNeil 1982; Fawcett 2006), as shown in the left panel of Figure 2.17, reproduced from Fawcett (2006). A perfect classifier would have a TPR of 1 and an FPR of 0, so it would have a coordinate of (0,1) on an ROC curve. The worst case would be if these values were inverted at (1,0). A random classifier would have these values be equal, and lie along the positive diagonal TPR = FPR. If a given number of classifiers are plotted on these axes then the one closest to (0,1) can be selected to maximise the recall and minimise the FPR. For algorithms such as the autoencoders used here, a threshold value is used to determine the classification. By varying the threshold value a curve may be plotted on these axes. When the threshold is set to exclude all inputs then it will be at (0,0) as no positives are recalled, but no negatives are falsely classified either. If the threshold is set to include all sources then it will be at (1,1) as all positives are recalled but all negatives are incorrectly classified positive as well.

The Area Under the Curve (AUC) beneath this curve from (0,0) to (1,1) is a metric of performance (Hanley and McNeil 1982; Fawcett 2006). As the curve cannot go below 0 or above 1 the AUC must lie between 0 and 1. If the classifier is perfect and goes to (0,1), then the curve fills in the whole square and AUC = 1. If the classifier is no better than random, then it lies along the diagonal TPR = FPR which gives AUC = 0.5. AUC = 0 would indicate a classifier that always returns precisely the wrong classification. The AUC is equivalent to the probability that a randomly selected positive will be ranked higher than a randomly selected negative. So we want as large an AUC as possible, and to at least exceed 0.5. The right panel in Figure 2.17 illustrates how two AUCs might be compared for two models.

## 2.10 Ensembles

The performance of machine learning models can often be improved by ensembling multiple diverse models together. Some of the most common ensembling techniques are discussed below.

Bagging (Breiman 1994), short for Bootstrap Aggregating, is intended to reduce variance and improve stability in machine learning models. The technique creates new training data sets from the original training set, one for each model, by sampling the original training set. The sampling is with replacement, which means that the set is randomly sampled with no regards for previous sets or samples. This may result in certain data being duplicated in the new training sets. The models are then

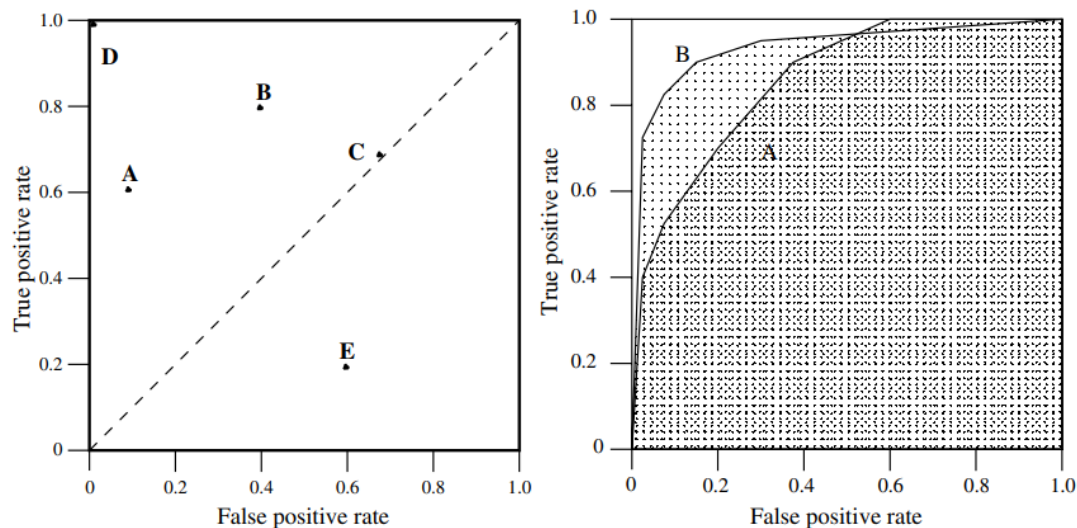


FIGURE 2.17: Two figures reproduced from Fawcett (2006). (Left) A basic ROC graph showing five examples of discrete classifier performance. **Classifier A:** Does reasonably well but is better at rejecting false positives than finding true positives. **Classifier B:** Also does reasonably well, but is more optimised to find true positives than reject false positives. **Classifier C:** No better than random. **Classifier D:** This classifier is perfect. **Classifier E:** This classifier does worse than random and is more likely to attach the wrong label than the right one. (Right) ROC curves for two classifiers with varying thresholds. The AUC is shaded for both. As can be seen, classifier B has a larger AUC and so is generally better than classifier A.

trained on these new sets. The models are combined to give final predictions by either averaging their predictions or by voting on the most common prediction in the case of classification.

Boosting algorithms attempt to create a strong classifier from several weaker ones. There are multiple boosting algorithms. One of the most common is AdaBoost (Freund and Schapire 1997). AdaBoost trains multiple models, attempting to correct the errors present in a model in the subsequent model. After each model is trained, AdaBoost assigns weights to the various training data. Those predicted more accurately by the model are given lower weights while those predicted incorrectly are given higher weights. When the subsequent model is trained higher emphasis is placed on those training data with higher weights in order to correct for the previous model. After each model is trained the model itself is also weighted according to its accuracy. The weighted average of these models is used to make predictions.

Stacking (Wolpert 1992), short for stacked generalisation, trains a model to combine the output of several other models. Stacking involves training multiple base models, often of different designs, on the complete training set. The models are then combined using a single meta-model. The way in which this is done is that the training data is split into  $k$  parts. For each of these parts the base models are trained on all  $k - 1$  other parts and then predictions are made for the withheld part. This is repeated for all parts. To train the meta-model, these predictions are then fed into the meta-model as features of the training data. The base models are then fit to the entire training set. The prediction on new or testing data is then obtained by first feeding

the input into the base models, and then the predictions of these base models into the meta-model which will make the final prediction.

Blending follows a very similar design to stacking and is most notable for being used in the winning solution to the Netflix Prize (Töscher and Jahrer 2009). The difference is that the blending technique withholds a validation set which is used to train the second layer model. From the training data some of the data are set aside for the validation set. The base models are trained on the training set without the validation data. The base models are then used to make predictions on the validation data. The predictions from the validation set only are used as features when training the meta-model.

## 2.11 Current Approaches to Classification and Anomaly Detection

Machine learning techniques, especially neural networks, have been widely and effectively used for anomaly detection (Hodge and Austin 2004; Chandola, Banerjee, and Kumar 2009), including anomaly detection in image data where the data has spatially correlated characteristics as well as some continuous attributes such as intensity or texture (Chandola, Banerjee, and Kumar 2009; Kwon et al. 2017; Nassif et al. 2021).

In radio astronomy, neural networks have been used for the classification of radio galaxies by morphology and to find anomalous sources. Aniyan and Thorat (2017), used CNNs to classify FRI, FR II and bent tail radio galaxies from the Very Large Array (VLA) Faint Images of the Radio Sky at Twenty centimetres (FIRST) survey. Around 100-200 sources of each class were augmented using rotations to 25000-36000 and used to train three binary classifiers, each distinguishing between a different pair of the three classes. Together they were taken as a fusion classifier, which combines the predictions of the three classifiers with their probabilities to make a final decision. For any of the three given classes, two will have been trained to distinguish it. If two classifiers agree on the classification with good probability then that classification is returned. If the probabilities are low or two cannot agree on the classification then it is marked as unusual or strange, potentially allowing for the discovery of exotic objects. The classifiers had good results comparable to manual classification. For FRIs, FR IIs and bent-tails precision values of 91%, 75% and 95% respectively were achieved with recall values of 91%, 91% and 79%, respectively. The work of Aniyan and Thorat is relevant as it concerns the classification of radio galaxies by morphology as well as some manner of ensembled CNNs. The data and application differ, however, in that there are only three morphologies of sources considered, classifiers are trained on highly augmented samples and the images used are substantially different from the MeerKAT images used here.

Other machine learning approaches have also been used to classify similar data. Mostert et al. (2021), used an unsupervised machine learning technique called Kohonen maps or Self-Organising Maps (SOMs, Kohonen 1988; Kohonen 2001) to classify around 25000 radio sources from the Low Frequency ARray (LOFAR, van Haarlem et al. 2013) Two-metre Sky Survey (LoTSS, Shimwell et al. 2017). SOMs work by creating a grid or map of cells and then organising the data into these cells such that

the sources in each cell are most similar to each other and cells neighbouring each other are the most similar. This is done by creating a prototypical image for each cell, initialised to be random noise or some selected initial images. The map is trained by iterating over the images and selecting the closest matching cell for each image. The selected cell is updated so that its prototypical image more closely resembles the input image, typically as a linear combination of its current form and the input, and neighbouring cells are likewise updated to a lesser degree with each cell being updated to an extent according to some drop-off function of its distance to the selected cell. Over several epochs the changes made to the map also drop off with increasing epochs. Images are then sorted into the cells based on the final map by finding the cell for each image that has the most similar prototypical image. The implementation of SOMs used was the Parallelised rotation/flipping INvariant Kohonen maps (PINK, Polsterer, Gieseke, and Igel 2015). In order to find rare or exotic sources, a threshold is used on the difference between an image and its closest matching cell. If the difference is large enough it is taken to be a rare source. Figure 2.18, reproduced from Mostert et al., shows the trained SOM (top) with each cell showing its prototypical image and highlighted to show a manually applied classification. The way a threshold may be applied to select the most unusual sources is shown in the middle. The bottom panel indicates how these Euclidean distances vary by the source type and how this may cause the threshold to have more false positives of certain types of sources and more false negatives of others. Although it is difficult to compare the performance of this approach directly, the way in which it handles different types of sources can be seen. It demonstrates that anomaly detection methods might favour certain source morphologies as anomalies more than others.

Autoencoders, including convolutional autoencoders, have been compared for anomaly detection by Doorenbos et al. (2021), to other anomaly detection methods. Doorenbos et al. (2021) compared a number of various machine learning approaches to outlier detection on the Sloan Digital Sky Survey (SDSS) Data Release 9 (York et al. 2000; Abazajian et al. 2009). This uses optical imaging data, so not directly comparable to this thesis, but may give some indication of how the algorithms compare. Six algorithms were compared. The first is Local Outlier Factor (LOF, Breunig et al. 2000) which detects anomalies by the "density" of the sources in local feature space, being the typical distance of a source's  $k$  nearest neighbours. Outlier sources are those with substantially lower density than their neighbours. Second, Isolation Forests (IF, Liu, Ting, and Zhou 2008) detect anomalies using decision trees which represent recursive partitioning of the data by a randomly selected threshold on a randomly selected attribute at each node in the tree. Each tree branches until every data point is isolated, called isolation trees. The number of branches required for a particular source to be isolated is its path length for that tree. Anomalies have more unusual features so will typically become isolated more quickly and have shorter average path lengths across a forest of isolation trees. Third, K-Means clustering (KM, Lloyd 1982) partitions the data into  $k$  clusters which minimise the variance within each cluster. Each cluster is represented by a mean of the data in that cluster and each object belongs to the cluster with the closest mean. The means are typically found by iterating over two steps, the assignment step, which assigns each object to the cluster with the nearest mean, and the update step, which updates each mean to be the mean of all objects in the cluster. The means are initialised either by randomly selecting  $k$  objects to be the initial means or by randomly assigning all objects to clusters and then proceeding with the update step. More efficient implementations are often used in practice. Fourth, Modified Novelty measure (MN, Hajer et al.

(2020) is a recently proposed variation on LOF that tries to take clustering of the data into account. It uses a new measure of novelty in which the dimension of the feature space is used and is proportional to the ratio of number of anomalous objects to the square root of the number of typical objects in a local volume. Finally, both normal and convolutional autoencoders are used.

For all algorithms, except the autoencoders, the set of images is first reduced using Principal Component Analysis (PCA, Pearson 1901; Hotelling 1933) so that each image is 14 components. PCA is a dimensionality reduction technique that attempts to maintain as much variance between the projected data points as possible. The principal components of a collection of points are a sequence of unit vectors, each the direction of the line that best fits the points while also being orthogonal to any previous vectors. PCA reduces the dimension of some higher dimensional data down to  $n$  by computing the first  $n$  principal components of the data and then using them as a orthogonal basis onto which the points are projected. It can be shown that the principal components are the eigenvectors of the data's covariance matrix, so this is typically how they are calculated. Doorenbos et al. used the algorithms to search the same data and extract what each considered to be outliers. The authors only compared the outliers produced by the algorithms with each other, to determine the similarity of the algorithms' output, and not with any catalogue of known exotic sources. So while it cannot be used to determine which algorithm had the best performance in detecting unusual sources, it can give an indication of which algorithms can be expected to have similar predictive outcomes. They found that convolutional autoencoders and LOF selected outliers most different to the rest of the algorithms which found outliers quite similar to each other. This can be some evidence that these algorithms are worth further exploration as they might produce quite different results which may be of interest.

As the goal is to locate anomalies rather than to produce a complete catalogue of reliable classifications, it may be desirable to optimise recall rather than precision. This means the user will need to examine more sources, although significantly fewer than searching the entire catalogue. Astronomy (Lochner and Bassett 2021) incorporates user feedback during the training process to improve results, called active learning. First, feature extraction algorithms, selected based on the type of the data, are run on the data, such as wavelet decomposition or ellipse-fitting of morphology features. Further post-processing, also included in Astronomy, may be run on the result of the feature extraction, such as feature scaling or PCA. An anomaly detection algorithm is run on the extracted features after post-processing which assigns an anomaly score to each source. Astronomy has two unsupervised anomaly detection algorithms, IF and LOF. The anomaly detection results are used to build a list of sources ranked according to their anomaly score. Some are shown to the user to provide a user score on how relevant the source is to their particular scientific interests. The user input is fed into another machine learning model called a regressor which calculates relevance scores for the unseen sources from the user input based on their similarity to the seen sources. A combined score is created and used to rank the sources based on the previous anomaly score and the regressor relevance score, weighted such that sources most similar to seen sources will have their combined score dominated by the regressor score, while those least similar will have their combined score dominated by the previously calculated anomaly score. The paper reports two performance metrics. Recall, and the Rank Weighted Score (RWS), proposed in Roberts, Bassett, and Lochner (2020), which assigns a score

based on how highly anomalies in the top  $N$  sources of the list are ranked. A higher RWS indicates anomalies in the top  $N$  ranked sources are closer to the top, while a lower score would indicate they are ranked lower, even for the same number of recalled anomalous sources. A RWS of 1 would indicate all top  $N$  sources are anomalies while 0 would indicate that none are. Figure 2.19, reproduced from Lochner and Bassett (2021), shows results of `Astronomy` run on optical Galaxy Zoo images (Lintott et al. 2008; Lintott et al. 2010; Willett et al. 2013). Looking at Figure 2.19 (a) we see that the false positive rate is fairly high in order to get a high recall. However, the user needs to look through far fewer sources in the machine ranked list to find a certain number of sources of interest than with the randomly ordered list. Trading recall for precision is useful here. As can be seen, factoring in the user feedback provides a performance increase to the list ordered without active learning. As this is optical data no direct comparison to our study is possible but a broader insight of `Astronomy` performance might be gained.

CNNs have also been used to search radio galaxy images for different types of morphological anomalies. Tang et al. (2022), looked at the use of deep learning CNNs in classifying Giant Radio Galaxies (GRGs, defined to be those radio galaxies with a projected linear size of over 700 kpc) in Radio Galaxy Zoo data (Banfield et al. 2015). Several architectures were tried on several combinations of data from the surveys FIRST, NRAO VLA Sky Survey (NVSS, Condon et al. 1998) and for some of them the redshift,  $z$ . The data consisted of selected radio galaxies with projected linear sizes of less than 500 kpc for typical sources and greater than 700 kpc for GRGs. For finding GRGs, they produced  $F_1$  scores varying between  $\sim 0.6$ - $0.8$ . Although the goal was to find very different types of sources than this project and the architecture of the machine learning algorithms used was quite different, it shows the applicability of using CNNs to search for anomalous radio galaxies by morphology. Not only can the image data being searched produce very different results, but so can the definition of anomalous sources being searched for.

Autoencoders have also achieved success in anomaly detection in image and other data outside of astronomy (Ding et al. 2019; Lu et al. 2017; Xu et al. 2015). A common method is that the autoencoders are trained on typical samples. Once trained, typical samples then result in a low reconstruction error while anomalous samples have higher reconstruction errors. This method and various modified versions of this method have achieved quite successful anomaly detection (Hasan et al. 2016; Zong et al. 2018), including on image and video data (Zhao et al. 2017; Gong et al. 2019). The success of autoencoders in anomaly detection makes them a promising choice for this project.

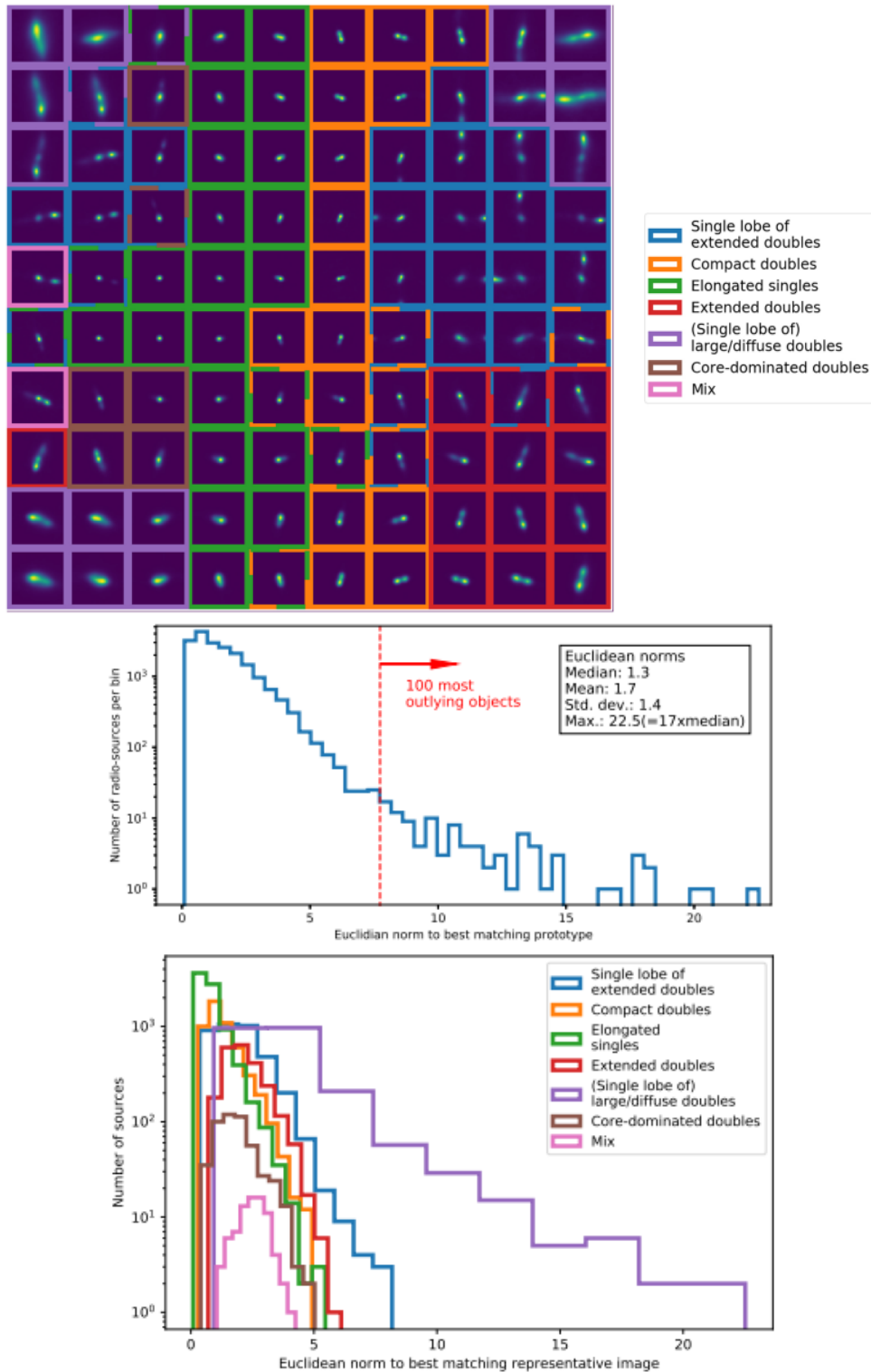


FIGURE 2.18: Three figures reproduced from Mostert et al. (2021) using LOFAR LoTSS images. (Top) A trained SOM showing the representative images of each cell and highlighted by a manually applied classification. (Middle) A histogram of the Euclidean norms to the best matching cells for the sources. For each source, this indicates the "distance" or difference between the source and the closest matching cell. Indicated is an arbitrary threshold, set to select the 100 most unusual objects. (Bottom) Euclidean norms of the sources to their best matching cells, separated by the classifications shown above.

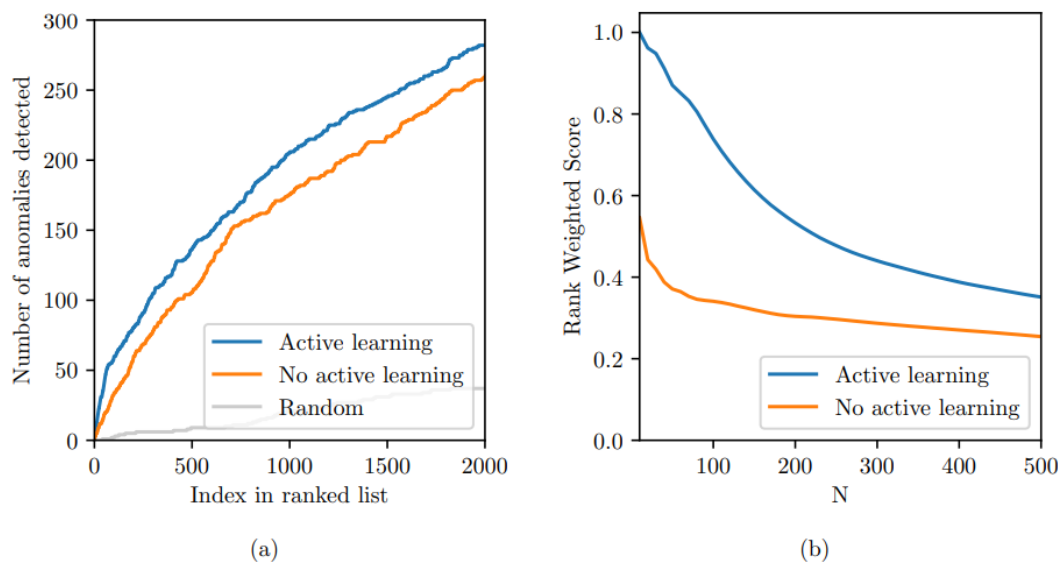


FIGURE 2.19: Reproduced from Lochner and Bassett (2021), the results of Astronomy using Galaxy Zoo data. True anomalous sources were taken to be those which more than 90% of human volunteers marked "odd." (a) For the sources ranked according to how anomalous they are predicted to be, the number of truly anomalous sources recovered by a certain index is shown for different methods of ranking. Plotted are the list ranked randomly, the list ranked using just the first model of machine learning, and the list ranked using active learning with both models. (b) The Rank Weighted Score for the top  $N$  sources on the ranked list.



## Chapter 3

# Methods

In this chapter we look at the specific approaches that are explored in this project. First is examined how the image data is processed and prepared for use in the autoencoders. Secondly the autoencoders themselves and their training. Finally, the performance evaluation of the various models is examined.

### 3.1 Data Processing

A subset of the galaxy cluster images were processed with the objectives of finding and making cutouts of extended radio sources while also minimising the number of point-like sources. The galaxy cluster images were processed using the source-finding software PyBDSF (Mohan and Rafferty 2015), with settings tuned to select for extended sources. The settings were chosen based on past experience with similar MeerKAT maps as well as some interactive tests. PyBDSF searches for and models sources in the larger image with ellipses. First, PyBDSF searches for image peaks that have a brightness or value sufficiently higher than the mean which are taken to be part of sources. This parameter, called *"thresh\_pix"*, is set to 30, so PyBDSF would detect any part of the image at least  $30\sigma$  above the mean as the peak of an "island" of pixels that form a source. This "island" forms the complete source and is composed of continuous pixels that are brighter than the background by a certain factor and have at least some pixels that have a peak brightness as described above. PyBDSF looks for the extent of the source around this peak using the lower  $\sigma$  threshold setting for the continuous pixels surrounding the peak. This allows it to model less bright diffuse emission surrounding the bright parts of the source and determines a border of the "island." If this setting is too low, background noise may begin being taken as part of the source. If it is too high then some of the emission will be cut off and not detected as part of the source. This threshold, called *"thresh\_isl,"* is set to  $5\sigma$ . PyBDSF will model what it finds by fitting Gaussians to it. It can output the results as either the individual Gaussian information, which may include point sources, or by island so care must be taken to use the correct output. It must be set to group the Gaussians together into islands by setting the *"group\_by\_isl"* option to true and disabling the *"split\_isl"* option which splits apart sources it thinks are too large. The tolerance for grouping these into islands should be set higher, using the option called *"group\_tol,"* to ensure that it will group the larger sources correctly. A value of 3 was used over the default of 1.

Many methods were tried in order to clean up the background and point sources. This includes using PyBDSF to subtract out the point sources, although it lacked the precision to do so well, often leaving point sources in the cutouts or removing parts of more diffuse, non-point sources. A similar attempt to subtract the background in an automated way was also unsuccessful. Although the residual of the PyBDSF

model did occasionally provide useful information during manual classification. So the residual was shown during that process along with the original image. The best results were found by simply cutting out the sources in squares as close as is possible in an automated way with side-length based on the size estimation returned by PyBDSF for each island. The background and blank pixels are filled with smoothed normalised random noise scaled to the residual background.

Next the correct classifications for the sources are required so that the algorithms may be tested against them as well as used for training. These images are manually classified by three people into several possible categories which were FRI, FR II, bent sources, S/Z shaped sources, sources with exotic morphologies as well as cluster emission. This is also used to flag images which have cutouts that are required to be removed as they need adjustment or are point sources. Multiple tags can be applied to any source and some sources fall into multiple categories, such as a bent FR II. Any such classifications which two or more people agree upon are used to label the source. Additionally, the exotic sources are manually inspected together in order to verify their classification. All three researchers agreed on the classification of 58% of the sources, while the remainder received the classification agreed upon by two out of the three. S or Z shaped galaxies made up just 1.6% of the sources manually classified while 14% were classified as bent. The majority were FR II galaxies, making up 65% of the sources classified, while FR I's were just under 20%. These may differ somewhat from the expected statistics due to the selection process.

Before training, the data is also augmented by rotating images 90 degrees as well as flipping both top to bottom and left to right, as shown in Figure 3.1. This type of augmentation should not affect the classification as the rotation of radio sources in the sky relative to observing telescopes should be random and have nothing to do with the morphology or intrinsic properties of the source. Augmentation increases the number of sources available for training and testing in order to provide a more thorough test. This increased the number of sources available for testing and training by four times for a total of 3700, of which 104 are exotic. 3327 of the typical sources are taken for training. The whole process can be seen in the diagram in Figure 3.2.

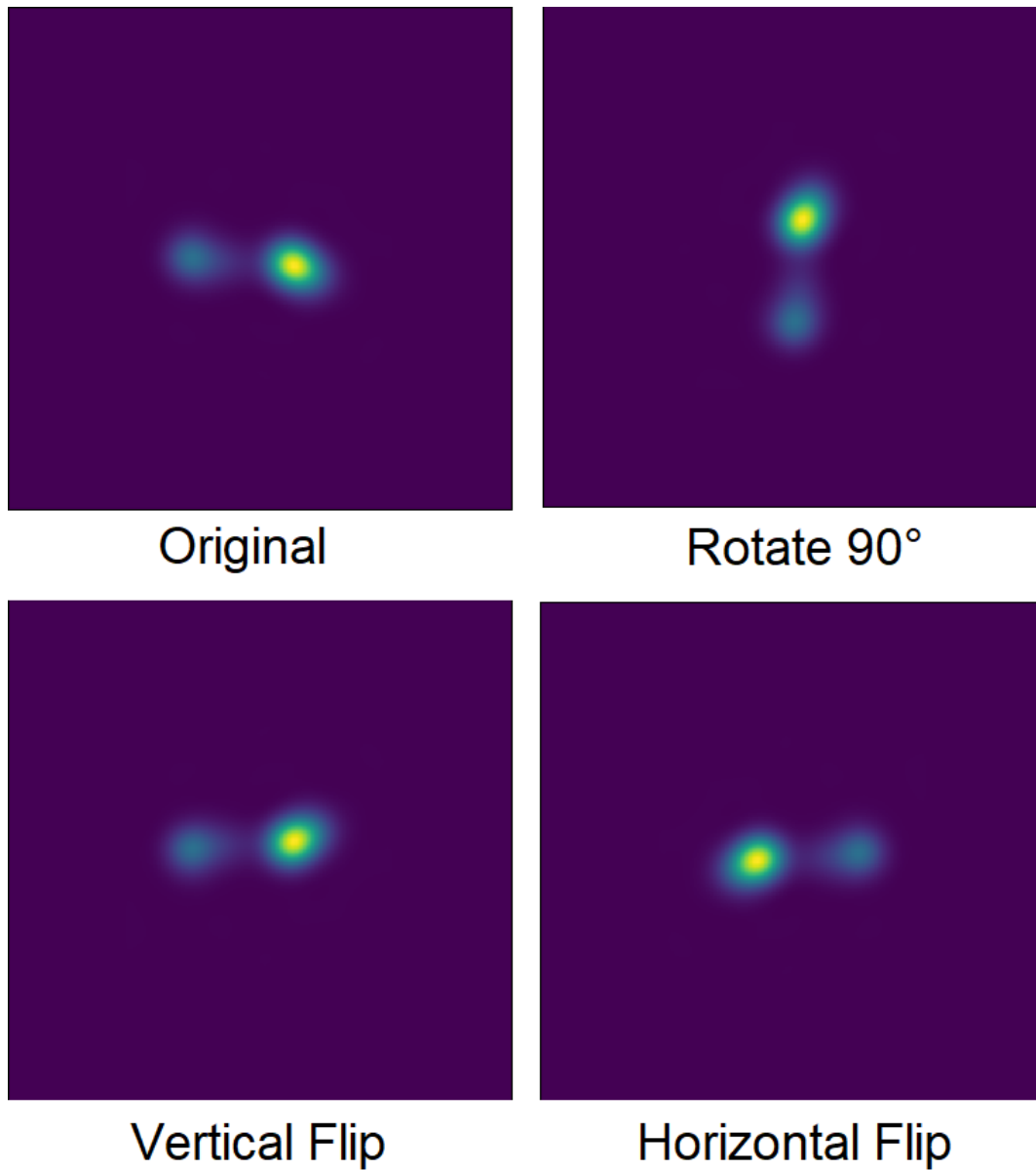


FIGURE 3.1: An example of augmentations applied to an image. (Top Left) The original image. (Top Right) The image rotated through 90 deg. (Bottom Left) The image flipped top to bottom. (Bottom Right) The image flipped left to right.

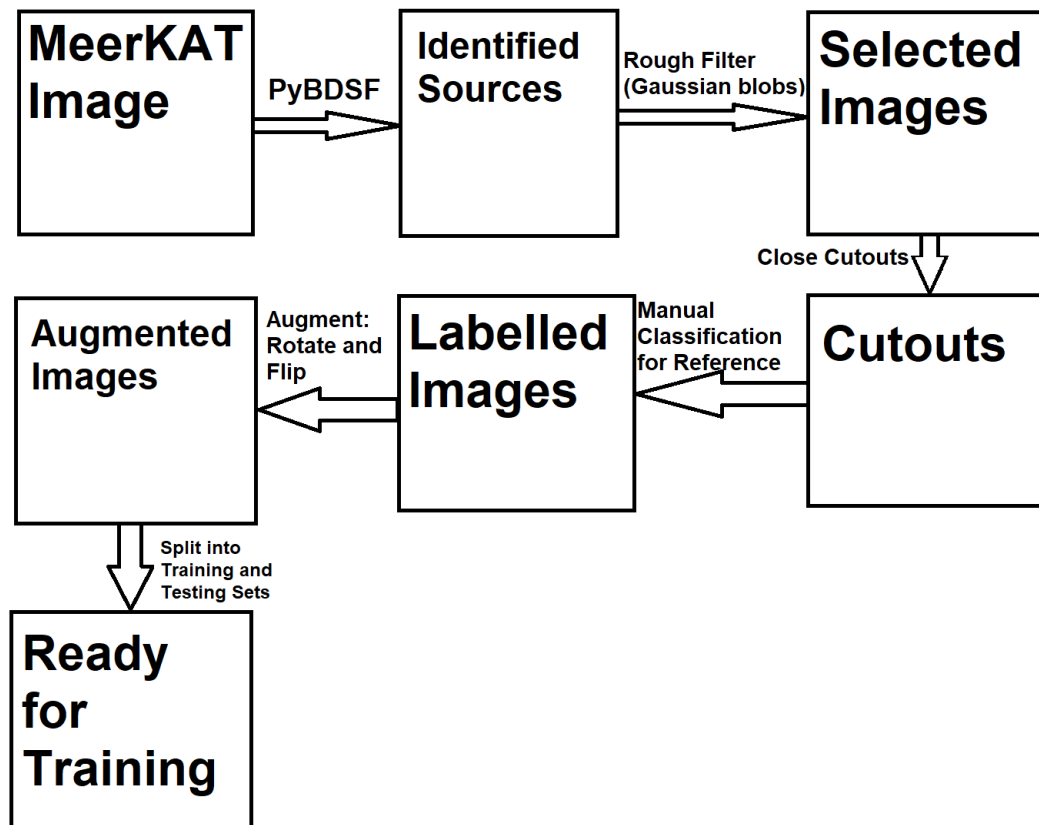


FIGURE 3.2: The process of making cutouts of the sources in the MG-CLS images. The cluster images were processed with PyBDSF which identifies the locations and estimates the angular extent of sources in the images. Close cutouts of the images were made and then they were manually labelled for testing. The images were finally augmented to increase the training sample size.

## 3.2 Autoencoders

A number of different arrangements of autoencoders are built using the software packages Keras (Chollet et al. 2015) and Astropy (Astropy Collaboration et al. 2013; Astropy Collaboration et al. 2018). The autoencoders all share the same basic structure, as shown in Figure 3.3. There is an encoder consisting of a SPP layer, followed by three 2D convolutional layers with 16 filters and  $3 \times 3$  kernels and finally a 2D max-pooling layer. The amount it is pooled controls the final size of the latent space and will differ for each setup. There is also the decoder consisting of a 2D upsampling layer that upsamples by the same factor as the max pooling layer reduces, followed by three 2D convolutional layers and a final convolutional layer that convolves to the same format as the input image. The two together make up the autoencoder with input going into the encoder, what it produces, being the encoded form, going into the decoder and the decoder output being taken as the output. They are compiled using the Adam optimiser implemented in Keras, a stochastic gradient descent method, with a cross entropy loss function and using accuracy as the metric. When autoencoders are trained the expected output is the same as the input.

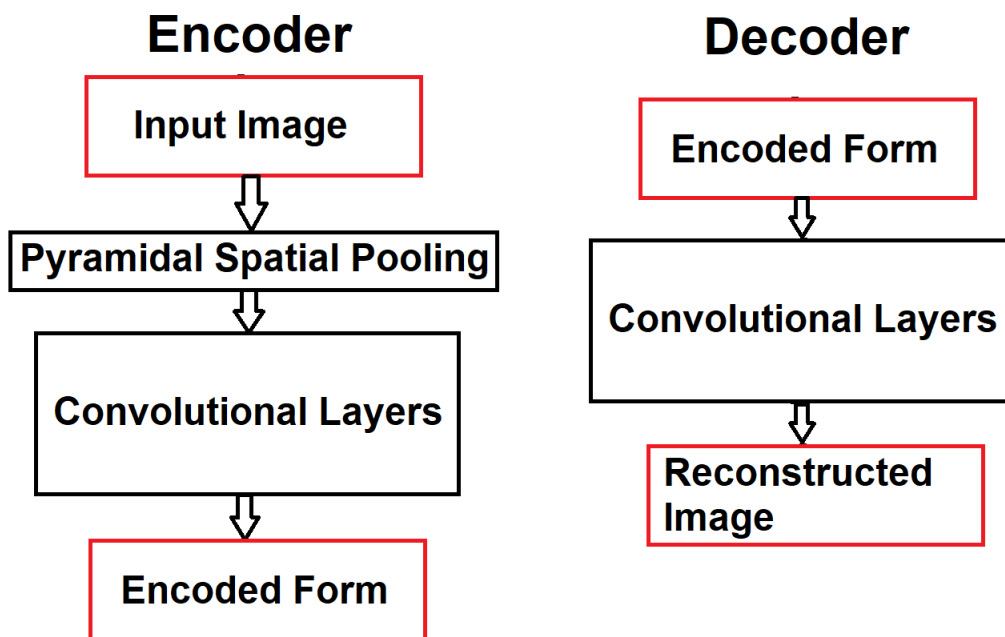


FIGURE 3.3: An autoencoder of the type that is used in this project consisting of convolutional layers and making use of a pyramidal spatial pooling layer with a max pooling layer just before the encoded form. The encoder CNN will take the input images and encode them onto the latent space while the decoder CNN will attempt to restore the original input image from the encoded form.

## 3.3 Ensembles

Various configurations or even initialisation of the models may have differing levels of accuracy with different types of sources or specific individual sources. This

variance can depend on the structure of the autoencoders as well as their initialisation. An autoencoder may reconstruct some typical sources well while struggling with others while another autoencoder may be better at reconstructing certain exotic sources. Typically, multiple models will not have difficulty with the same sources. By using multiple configurations of the model together it may be possible to significantly improve the classification results by eliminating these differences in reconstructing various sources.

Multiple configurations of ensembling the autoencoder based models are tried. The output of multiple encoders is simply averaged or voted on, similar to bagging, rather than the use of a meta-model such as is found in stacking. A reconstruction metric of the image is considered against a threshold in two ways. The first is to average the reconstruction accuracy and test this averaged value against a threshold in order to test if a source is exotic or not, which we will call averaging. The second is to test each reconstruction accuracy against the same threshold separately and then to take the classification agreed upon by the majority of the autoencoders, which we will call voting. All of the models are trained on the same data, and differences between the trained models are due to differences in model design and random initialisation. In this project approaches such as true bagging would have been difficult due to the limited number of sources available for training.

The total number of and variations between the structures of the autoencoders are also tested. An ensemble of autoencoders with the same structure but different weight initialisations, an ensemble of autoencoders with varying latent spaces, such as the one shown in Figure 3.4, and an ensemble with both multiple encoders of the same latent space as well as varying latent spaces.

The voting method will only classify a source as exotic if the majority of the autoencoders struggle to reconstruct it with an accuracy above a certain threshold, no matter how well one or the other does. This accounts for the possibility that some configuration may do well at reconstructing certain exotic sources by chance, even if it hasn't been trained on them, if certain features that it has learnt are common with the exotic sources. However it is unlikely to be true of multiple configurations of autoencoders. On the other hand it assumes that most of the autoencoders will all perform well with the types of sources that they have been trained on.

The averaging method will have the the autoencoders reconstruct the source and then take the average of the reconstruction score and test that against the threshold to determine the classification. This does mean that if a source is close to the threshold a single autoencoder with an abnormal performance may put it on the wrong side of the threshold.

The various setups used are show in Table 3.1. The first setup consists of a single autoencoder with a latent space of  $16 \times 16 \times 16$ . This will allow for the evaluation of a single autoencoder against the ensembles. Second is an ensemble of three autoencoders with the same  $16 \times 16 \times 16$  latent space as before that should ideally have minor differences in their results as a result of the initialisation. The output from the autoencoders will be ensembled together using two methods, the voting and averaging methods. Third an ensemble of three autoencoders but with differing latent spaces of  $32 \times 32 \times 16$ ,  $16 \times 16 \times 16$  and  $8 \times 8 \times 16$  respectively. The differences in performance should be more marked here, and the hope is that the way in which

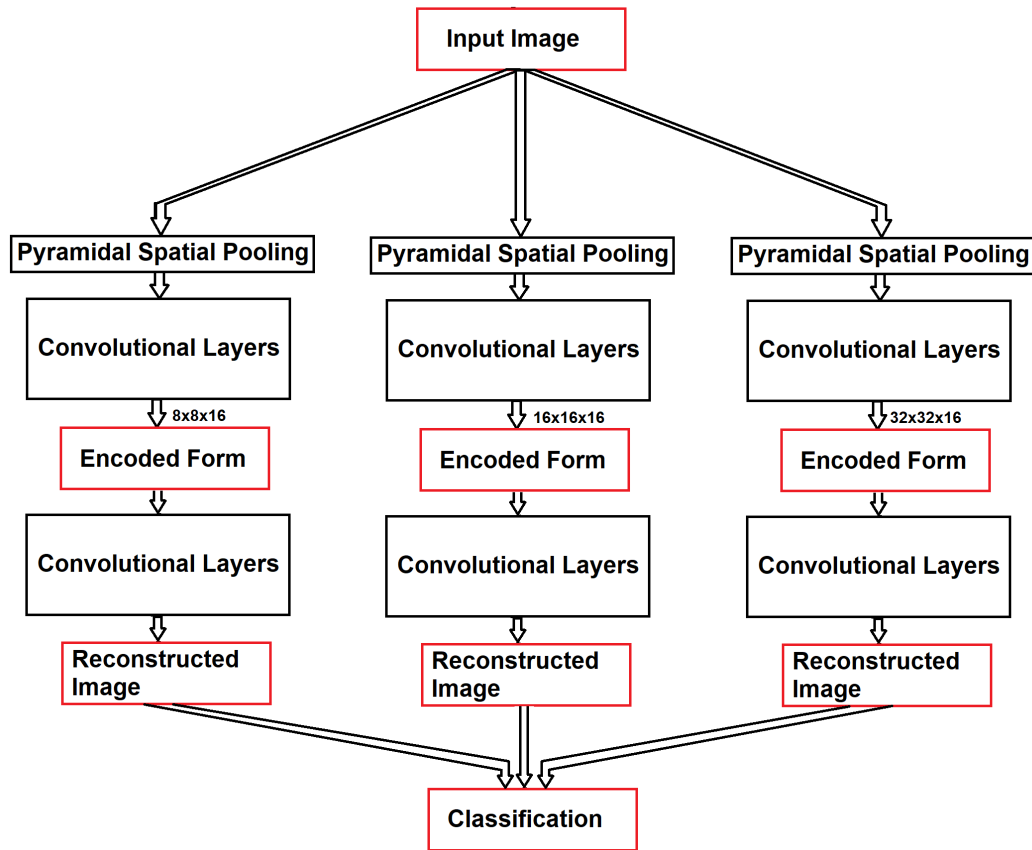


FIGURE 3.4: Diagram showing three typical autoencoders working together in ensemble. Here they have three differing latent spaces although other arrangements may have autoencoders with the same latent space dimensions.

they learn to encode the input differs enough to ensure most exotic sources will not be well encoded by two or more of them, while they will have been trained to reconstruct the typical sources. Finally an ensemble of nine autoencoders as described above but with three having latent spaces of  $32 \times 32 \times 16$ , three having latent spaces of  $16 \times 16 \times 16$  and three having latent spaces of  $8 \times 8 \times 16$ .

TABLE 3.1: The various setups that are used in this project, showing both the number of autoencoders in the ensemble for each setup as well as their latent spaces.

Setup Number	Number of Models in Ensemble	Latent Space Dimensions
1	1	$1 \times (16 \times 16 \times 16)$
2	3	$3 \times (16 \times 16 \times 16)$
3	3	$1 \times (32 \times 32 \times 16), 1 \times (16 \times 16 \times 16), 1 \times (8 \times 8 \times 16)$
4	9	$3 \times (32 \times 32 \times 16), 3 \times (16 \times 16 \times 16), 3 \times (8 \times 8 \times 16)$

The autoencoders must be trained to reconstruct the typical sources as well as

possible. This involves updating the weights of the various layers so that the reconstruction of one of these images is as close as possible to the original input image. Given the restrictions on the encoded form, this training should ideally only work well for those typical images and fail when it comes to the exotic sources allowing them to be identified. Training is done on a set of the typical images set aside specifically for training and which will not be used in testing. The autoencoders are trained on the training data set over 50 epochs with a batch size of 32. For validation the autoencoders are trained and tested on randomly split datasets five times. The results are then averaged over the five runs for each model in order to get a more robust metric of the performance.

All are tested on the same data. For each run the data is randomly split into a set of 3327 typical sources for training, a set consisting of half the remaining typical sources and half the exotic sources for finding a suitable threshold and a set of the remaining exotic and typical sources for testing the  $F_1$  and  $F_2$  scores at those thresholds. This is repeated five times, with both the averaging and voting methods being evaluated, and the average  $F_1$  and  $F_2$  scores are then found. This type of statistical validation gives a better indication of the performance of the algorithms rather than the chance performance of a single run and avoids shot noise.

### 3.4 Performance Evaluation

In order to evaluate how well an autoencoder has reconstructed a particular image the input and output images are compared and the Normalised Cross-Correlation (NCC) is calculated. This is not used during training but rather to evaluate the reconstruction of images for classification purposes. A cross-correlation is a similarity measure that is very similar to convolution (Goodfellow, Bengio, and Courville 2016). For two two dimensional arrays,  $I$  and  $K$ , of  $m$  by  $n$  pixels, it is given by

$$(I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n). \quad (3.1)$$

A normalised cross-correlation is used in order to account for variations in intensity and background noise that should ideally not affect the outcome of the similarity measurement. It can be normalised, where for every step it is divided by the standard deviation of the two images being compared, or zero normalised in which the mean of each image is also subtracted. The NCC measure for two such images,  $f$  and  $t$ , with dimensions  $m \times n$  is

$$NCC = \frac{1}{mn} \sum_x \sum_y \frac{1}{\sigma_f \sigma_t} (f(x, y) - \mu_f)(t(x, y) - \mu_t), \quad (3.2)$$

where  $\sigma$  and  $\mu$  are the standard deviation and mean of each image, respectively. The value is averaged over all pixels and is equal to 1 only if the images are identical. The closer to 1 the value is the more similar we take the images to be.

For any image, the normalised cross correlation may be calculated between the input image and the image reconstructed by a trained autoencoder in order to determine the reconstruction success. An encoder trained only on typical sources is



expected to perform better at reconstructing typical sources and poorly at reconstructing the sources with exotic morphologies. By setting a threshold for the reconstruction accuracy, ideally the majority of typical sources will have better reconstructions than the threshold while the majority of exotic sources will have poorer reconstruction performance allowing the two populations to be separated.

Once they have been trained on the training set, the autoencoders are used to encode and then decode the testing images. For each of these testing images the NCC is calculated between the original image and this reconstructed image. Classification is done by applying a threshold value between 0 and 1 to the NCCs. Anything above the threshold is assumed to be well enough reconstructed to be similar enough to the training images which consisted of typical sources. Anything below that threshold is then labeled exotic. For those autoencoders that are in an ensemble the final result is taken in one of two ways. A given original image will have a different reconstruction and so a different NCC for each autoencoder in the ensemble. The "averaging" method, Figure 3.5 (Left), takes the average of the NCCs for a particular image across all the autoencoders in the ensemble and then tests this against the threshold to determine the classification. The "voting" method, Figure 3.5 (Right), tests the NCC for each autoencoder against the same threshold individually, and each will have its own classification. It then selects whichever classification was applied for a majority of the autoencoders.

Those with highest and the lowest NCC values are useful for manual inspection. It is also useful to compare the best and worst typical source reconstructions as well as the best and worst reconstructed exotic sources.

The ROC can then be plotted by varying the threshold between 0 and 1. As the goal is to locate exotic sources, the exotic sources may be considered positives while the typical sources may be considered negatives. At each threshold step the FPR and TPR are evaluated and plotted. The confusion matrix and  $F$  scores can also be calculated for a given threshold. For the testing in this project  $F_2$ , which assigns double the significance to recall, is evaluated in addition to  $F_1$ .

In order to determine the  $F_1$  and  $F_2$  scores the best threshold will need to be selected. In order to have a more reliable and validated metric of the performance of the autoencoders, the threshold cannot simply be selected so that it maximises these scores for a given run as there would be no way to determine those values beforehand if we were not dealing with a labelled testing set. So the sets of typical testing images and exotic testing images are both randomly split into two equal sets. One half is used to find the threshold by maximising it for that half. The performance is then evaluated using the other half. This is repeated over five runs with the training, testing and threshold-finding sets being randomly split each time. The average is taken over these runs for both the threshold and the resulting  $F_1$ ,  $F_2$  and confusion matrix values. It should be noted that the values for the  $F_1$  and  $F_2$  scores and the confusion matrices are averaged separately. The confusion matrices are important to look at to understand the way in which the sources would finally be distributed. The  $F_1$  and  $F_2$  scores are important for understanding the level of performance of the model both when weighting precision and recall equally and when prioritising recall. As they are averaged separately the averaged confusion matrix values might not yield the averaged  $F_1$  and  $F_2$  scores if the formula is applied to them directly.

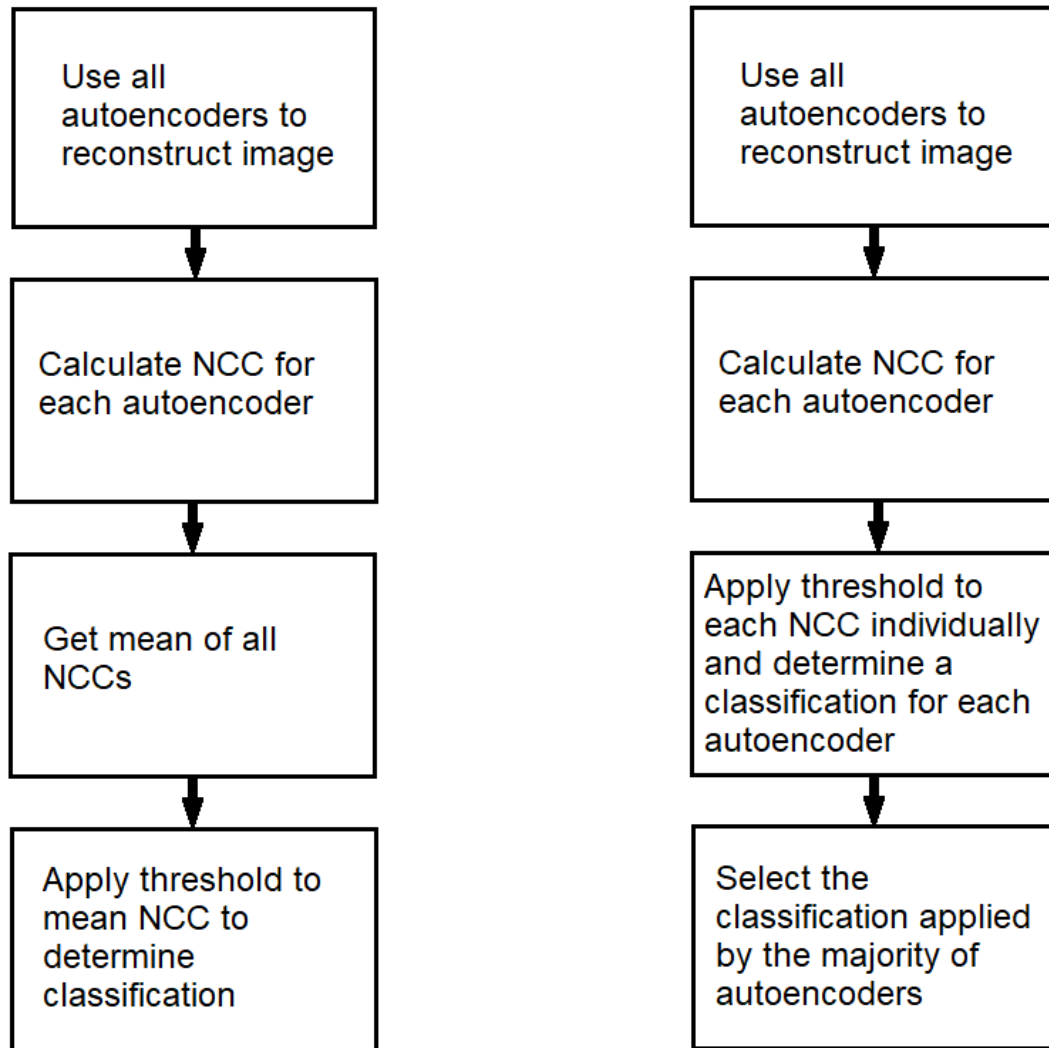


FIGURE 3.5: Summary of the methods for determining the classification of a single image using a given threshold. (Left) The averaging method, where the threshold is applied to the mean of the NCCs calculated between the original and reconstructed images for all models in the ensemble. (Right) The voting method, where the threshold is applied to each NCC for each model in the ensemble individually to acquire individual classifications, and the classification that most models agree upon is applied to the image.

As we are looking at both  $F_1$  and  $F_2$  and each of these may be optimised at different thresholds, different thresholds and confusion matrix values are kept for each. For the ensembles, as we are also looking at the difference between the averaging and voting ensembling methods, these too need to be kept separately. So for an ensemble there are four thresholds and results being the  $F_1$  voting values, the  $F_1$  averaging values, the  $F_2$  voting values and the  $F_2$  averaging values.

Although it will not be used in classification, it may give some insight to inspect the latent space for some of these autoencoders. PCA can be used to project the encoded latent space representations of the testing images onto a 2D plane. No restrictions were set on these latent space representations to try and get similar classes

to cluster together or other useful properties of the latent space representation, however, so they will not be used for classification.

## Chapter 4

# Results

In this chapter the results of running the various autoencoder ensemble models described in Chapter 2 on the data as described is presented. The results of each setup are presented separately and in the order described in Table 3.1.

### 4.1 Single Autoencoder

Data are passed through a single autoencoder with a latent space of  $16 \times 16 \times 16$ . For one run the testing images encoded in the latent space is analysed with PCA. The result is shown in Figure 4.1. Although there is some minor separation the two classes of objects cannot be readily distinguished in the latent space. This suggests that the two classes may not be very separable, although recall may still see an increase and there are regions without any exotic sources.

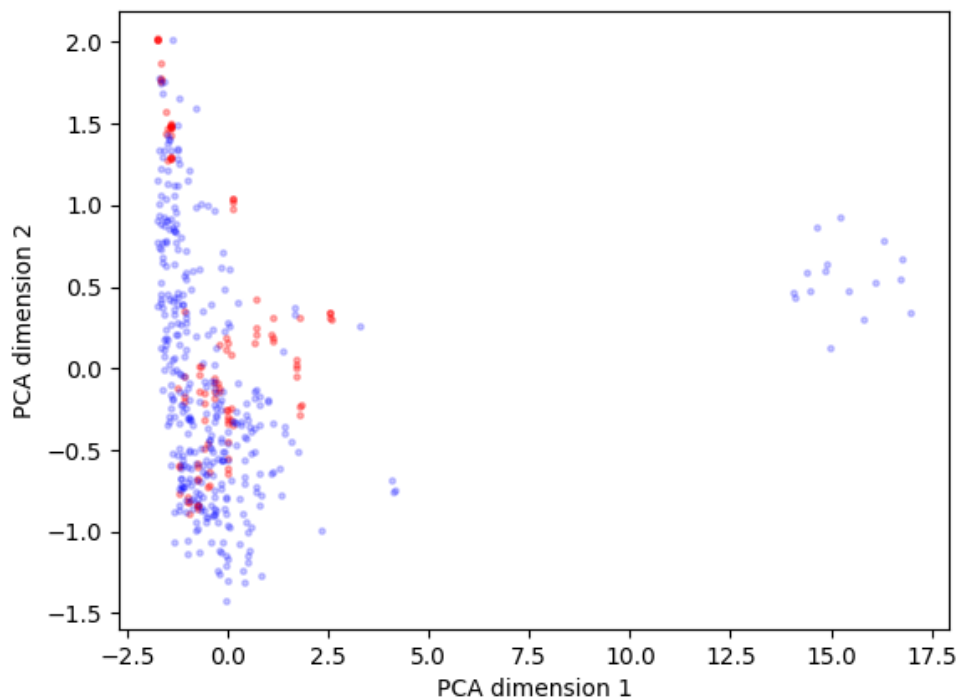


FIGURE 4.1: PCA output for the single autoencoder when run on the encoded output in latent space. The blue dots indicate typical testing sources while the red dots indicate exotic sources.

For the testing images the NCCs are then calculated between the original input images and the reconstructed images. Histograms of the distribution of these NCC values for the typical sources and the exotic sources are shown in Figure 4.2. There is an obvious difference in these distributions towards worse reconstructions but a significant overlap towards the higher end.

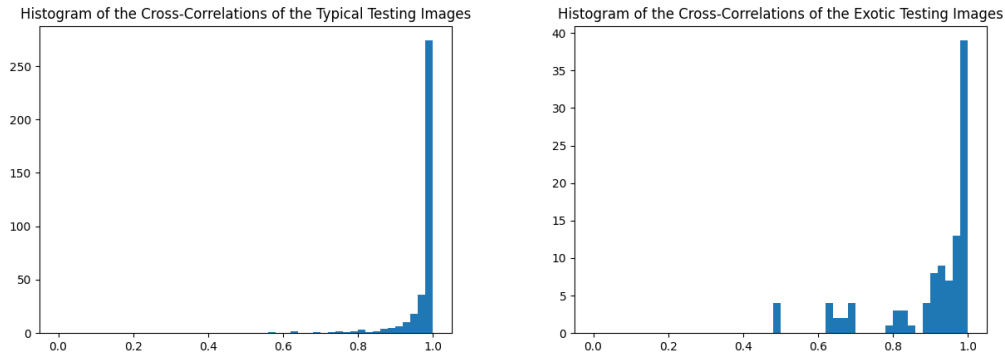


FIGURE 4.2: Instance of the results produced by the single autoencoder. (Left) Histogram showing the distribution of the NCC values for the typical testing sources. Lower values indicate a worse reconstruction while higher values indicate a better one. (Right) Histogram showing the NCC values for the exotic testing sources. It can be seen that more of these sources have worse reconstruction values although there is an overlap towards the higher end.

Figure 4.3 shows the ten best and worst reconstructed typical and exotic test sources. For each source it shows the original, the reconstruction and the residual. The residual is the subtraction between the original and reconstructed images and allows us to see what the algorithm has difficulty in reconstructing. As augmented images typically have relatively similar scores to their originals, as can be expected, they have been omitted and only the ten best and worst original images in each category is shown. For the best reconstructed images we can see that the residual typically has very low SNR emission remaining, which is expected. It can also be seen that the residuals for the exotic images appear to be more pronounced than those of the typical images, which is also expected. The worst reconstructions have much larger residuals that show that many of the images could not be accurately reconstructed. Looking at the images that it has difficulty reconstructing accurately, we see it struggles with low signal to noise images. The best reconstructed exotics seem to resemble typical radio galaxies more closely with lobes and similar structure while the worst reconstructed exotics seem to be sources such as relics with little structure resembling anything in the training data. This is to be expected as the autoencoders will be best at reconstructing whatever more closely resembles the typical source morphology encountered in the training data.

The ROC curve for the run is shown in Figure 4.4. A AUC of 0.76 is achieved. This is substantially better than the 0.5 of randomly ordered sources and so it can be seen that the model is at least somewhat effective.

In order to get a more accurate statistical measure of the performance, the algorithm is run five times, as described in Section 3.4. Each time the data is randomly

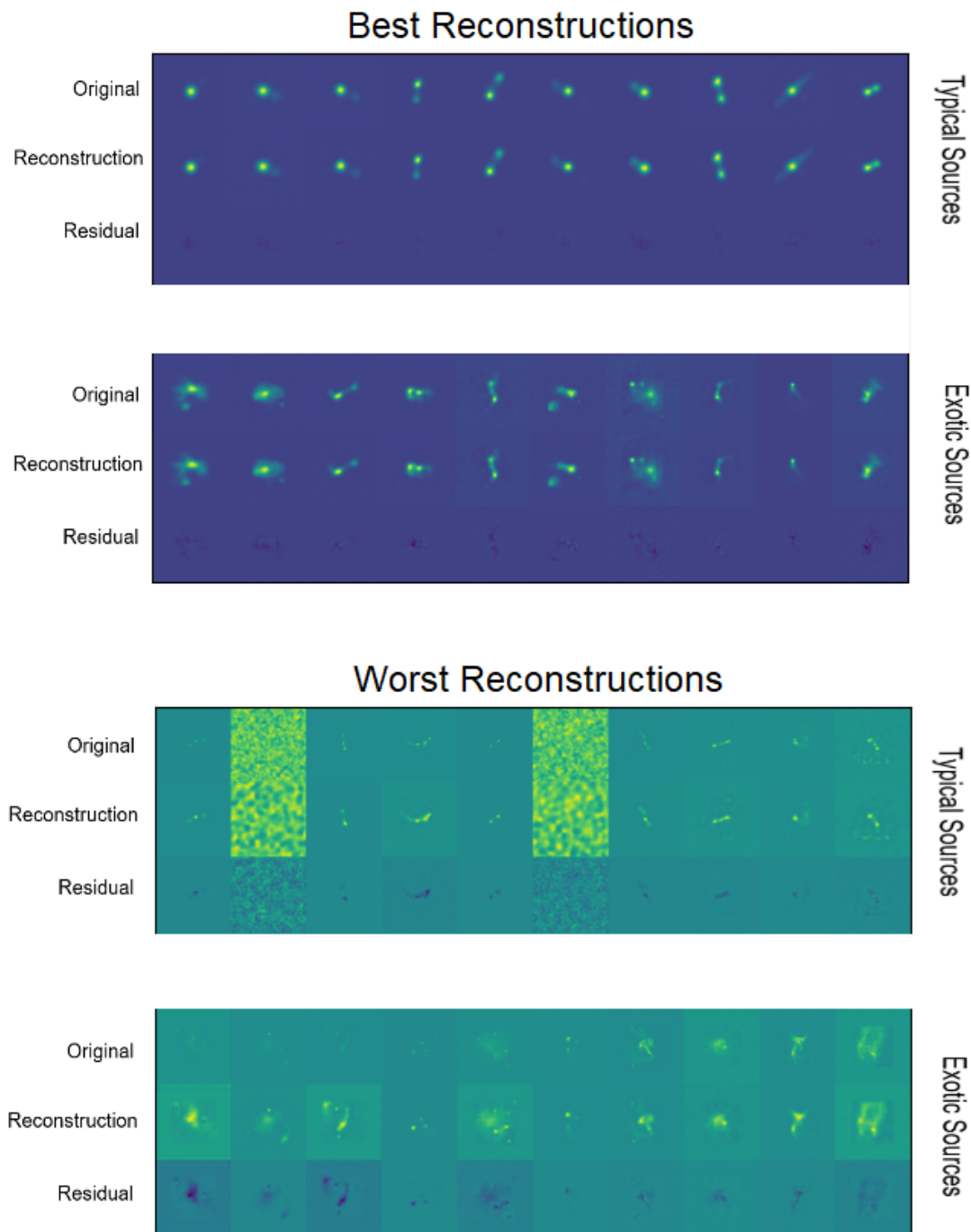


FIGURE 4.3: The ten best and worst reconstructed sources from both the exotic and typical test sources produced by an instance of the single autoencoder. (Top) The ten best reconstructed exotic and ten best reconstructed typical images from the testing data. For each image the original, the reconstruction and the residual is shown. (Bottom) The ten worst reconstructed exotic and ten worst reconstructed typical images from the testing data.

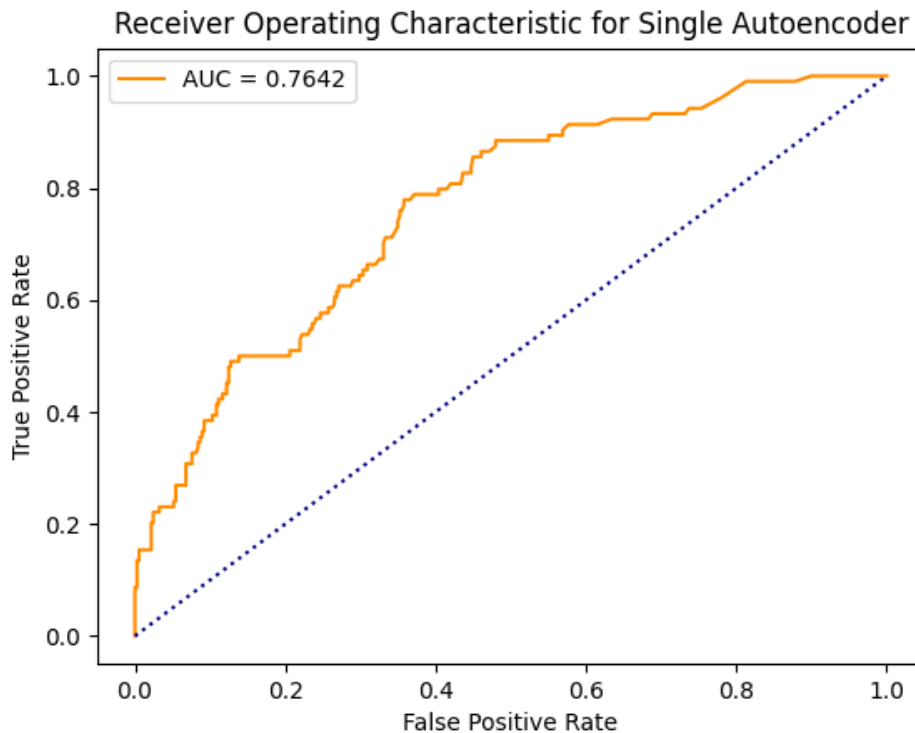


FIGURE 4.4: The Receiver Operating Characteristic curve for the single autoencoder.

split into the training and testing datasets. In order to get the  $F_1$  and  $F_2$  thresholds, half of the testing data is used to find the optimal threshold for each metric and the other half is then used to calculate the confusion matrix values and the score. Over all of the runs these values are then averaged to produce a more reliable estimate of the performance. The thresholds found this way were 0.9663 for the  $F_1$  score and 0.9864 for the  $F_2$  score. Figure 4.5 shows confusion matrices and  $F_1$  and  $F_2$  scores produced this way. Over the five runs the  $F_1$  score averaged 0.48 with a standard deviation of 0.02 while the  $F_2$  averaged 0.69 with a standard deviation of 0.01. The values in these confusion matrices do not necessarily sum to one as they are the average of each value over the five runs. It can be seen that when using the  $F_1$  threshold the FPR is high. However the  $F_2$  score is significantly better. It cannot be used to make reliable classifications as the precision only averages at 0.35, but a user searching for interesting sources will have more than half of the uninteresting sources removed while very few of the interesting sources will be removed along with these typical sources with an average recall of 0.92 having a standard deviation of 0.02 across the five runs. Using the threshold optimised for  $F_2$ , recovering many interesting exotic sources will require the user to look through far fewer sources. This is similar to the example from *Astronomy*, where many false positives remained but the number of sources required to look through to recover a certain number of anomalous sources was greatly reduced.

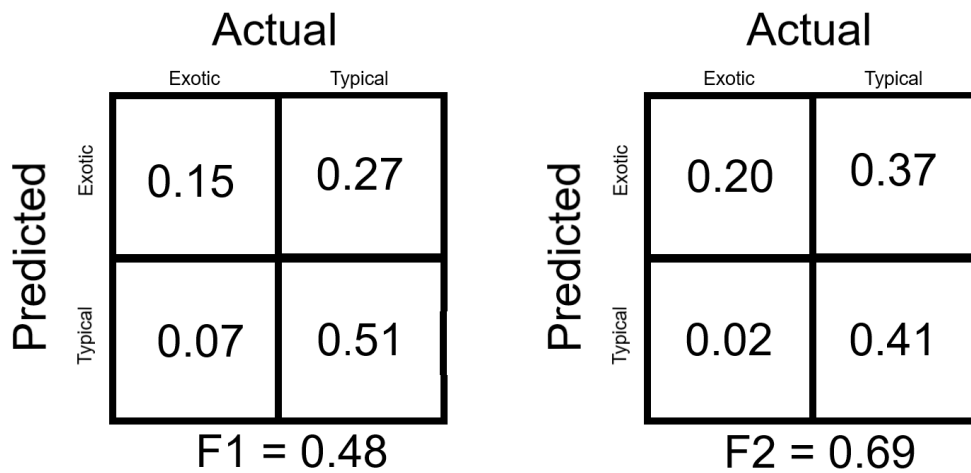


FIGURE 4.5: For the single autoencoder. (left) The confusion matrix and  $F_1$  score averaged over multiple runs of the algorithm for the threshold 0.9663. (Right) The confusion matrix and  $F_2$  score averaged over multiple runs of the algorithm for the threshold 0.9864.

## 4.2 Ensemble of Three Similar Autoencoders

Next, an ensemble of three autoencoders, having latent spaces of  $16 \times 16 \times 16$  similar to the previous one, are trained on the same data.

Histograms of the average NCC for each source across the three autoencoders are shown in Figure 4.6 for the typical and exotic testing images. Again we can see that, when compared with the typical sources, the exotic images have a larger fraction of sources towards the lower end of the spectrum.

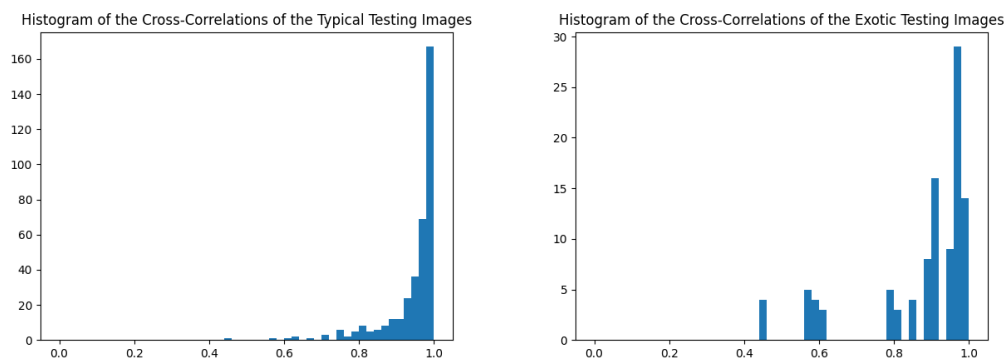


FIGURE 4.6: Example of the reconstruction abilities from an ensemble with three similar autoencoders (Left) Histogram of the distribution of the average NCC values of the typical testing images. (Right) Histogram of the average NCC values of the exotic testing images.

Two approaches are used to combine the scores from the three autoencoders, the averaging method and the voting method. For both methods, the ROC curves are shown in Figure 4.7. As can be seen here they produce rather similar results with AUC scores of around 0.7 for averaging and 0.73 for voting, although fluctuations



from run to run can put these in line with each other and the single autoencoder.

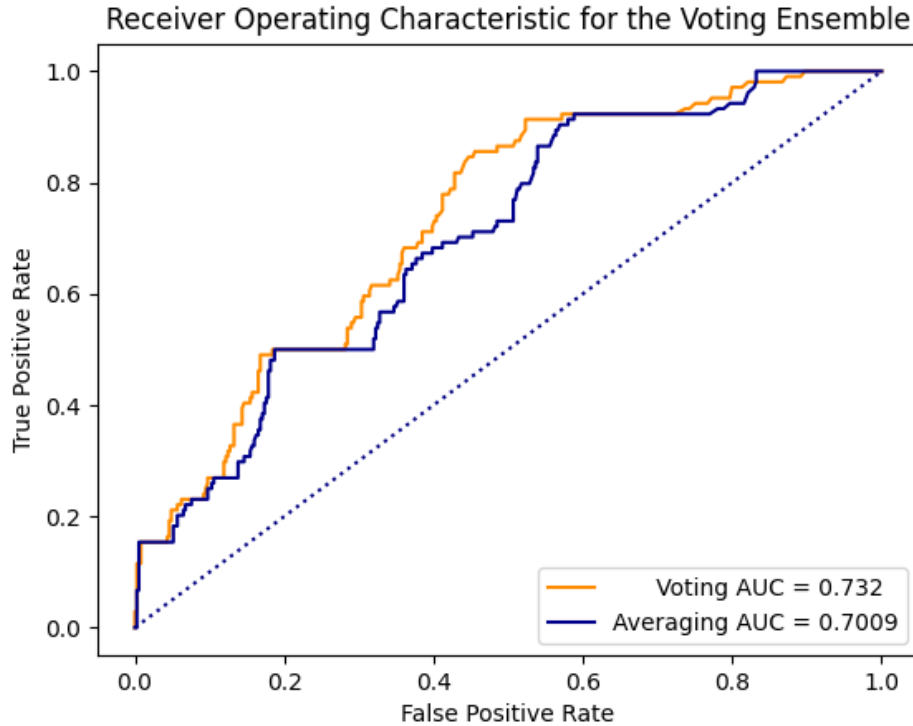


FIGURE 4.7: ROC curves for the ensemble of three similar autoencoders for both the averaging and voting methods.

As before, the thresholds optimised for  $F_1$  and  $F_2$  as well as the  $F_1$  and  $F_2$  scores themselves and the values in the confusion matrices are calculated over five runs with the training and testing data being randomly split in each run. For the ensemble they are calculated using both the averaging method and the voting method. In Figure 4.8 the confusion matrices are provided for the averaging method. Thresholds are 0.9764 and 0.9892 for  $F_1$  and  $F_2$ , respectively. Figure 4.9 shows the same for the voting method. Thresholds are 0.9780 and 0.9896 for  $F_1$  and  $F_2$ , respectively. Here it can be seen that the results between the voting and averaging methods for this particular ensemble are  $<0.1\%$  apart. Their performance is also comparable to the single autoencoder. The voting method produced an average  $F_1$  score of 0.5, with a standard deviation of 0.02 across the five runs, and an average  $F_2$  score of 0.67, with standard deviation 0.03. It also produces an average recall of 0.88, with a standard deviation of 0.07 across the five runs, at the  $F_2$  threshold and a recall of 0.76 with a standard deviation of 0.13 at the  $F_1$  threshold. These unusually high standard deviations suggest the run to run variance is still high despite the ensemble.

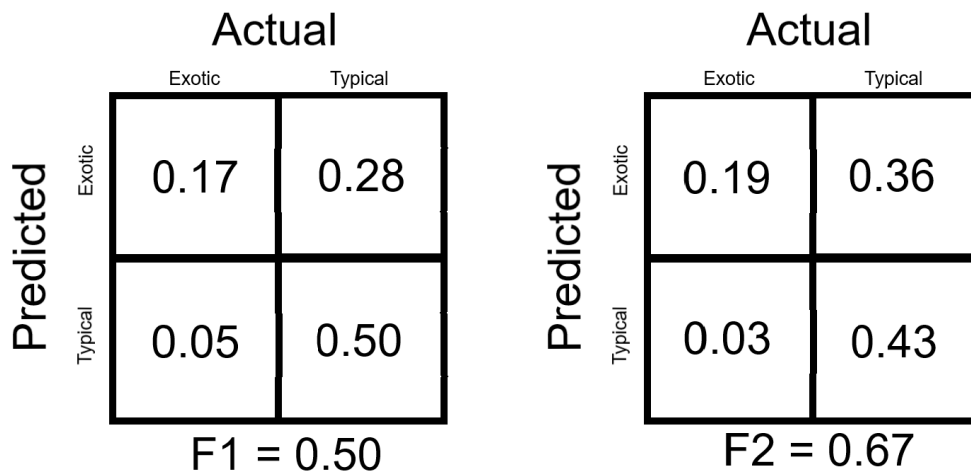


FIGURE 4.8: The  $F_1$  and  $F_2$  scores and confusion matrices for the ensemble of three similar autoencoders using the averaging method. (left) The confusion matrix and  $F_1$  score of the threshold optimised for  $F_1$ , 0.9764. (Right) The confusion matrix and  $F_2$  score of the threshold optimised for  $F_2$ , 0.9892.

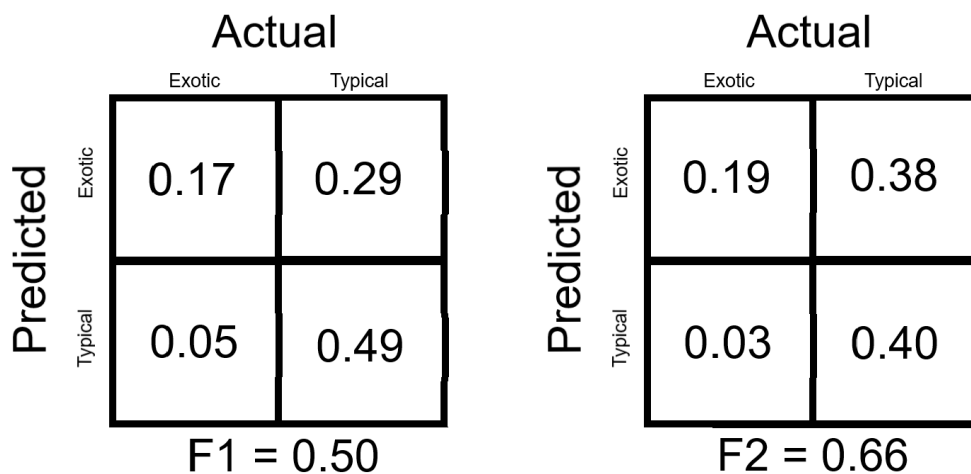


FIGURE 4.9: The  $F_1$  and  $F_2$  scores and confusion matrices for the ensemble of three similar autoencoders using the voting method. (Left) The confusion matrix and  $F_1$  score of the threshold optimised for  $F_1$ , 0.9780. (Right) The confusion matrix and  $F_2$  score of the threshold optimised for  $F_2$ , 0.9896.

### 4.3 Ensemble of Three Different Autoencoders

Another ensemble of three is also tested, this time with three differing encoders. The structure of the encoders is the same but they reduce the images down to different latent spaces of sizes  $8 \times 8 \times 32$ ,  $16 \times 16 \times 16$  and  $32 \times 32 \times 32$ .

PCA was run on the encoded forms of the testing data for the different latent spaces. Figure 4.10 shows the results. Some differences are noticeable between the way the encoded images have been mapped to 2D. Like the previous autoencoders, no constraints have been placed to encourage separation so the encoded form is not useful for classification here. The most noticeable differentiation appears to be from

the first autoencoder.

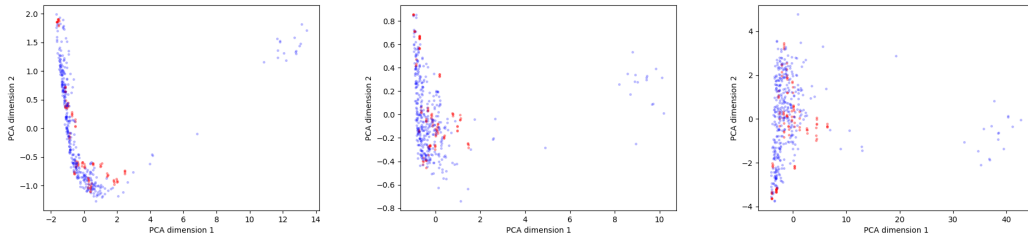


FIGURE 4.10: The encoded forms of the test sources in the three latent spaces of the varying autoencoders mapped into 2D using PCA. Red points indicate exotic test sources while blue dots indicate typical testing sources. (Left)  $16 \times 16 \times 16$  (middle)  $8 \times 8 \times 32$  (Right)  $32 \times 32 \times 32$ .

Histograms of the average NCCs of the typical and exotic test data across the three autoencoders for a run of the ensemble are shown in Figure 4.11. Here it already looks significantly different to the previous autoencoders, with both being skewed more broadly to the lower end.

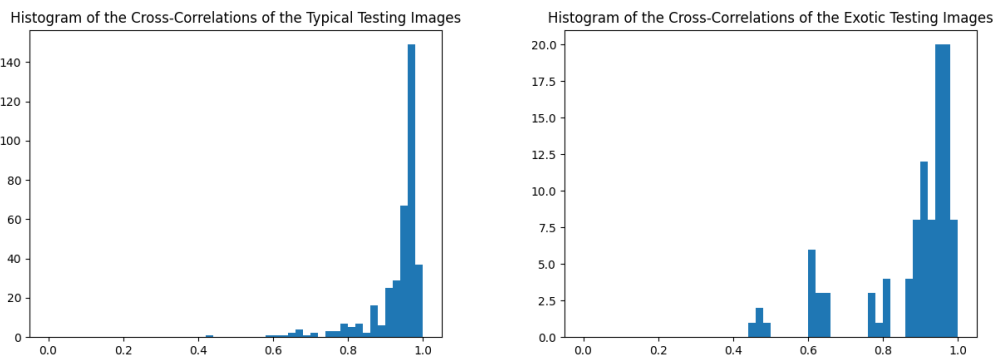


FIGURE 4.11: An example of the reconstruction ability of the ensemble of three differing autoencoders. (Left) Histogram of the distribution of the average NCC values of the typical testing images. (Right) Histogram of the average NCC values of the exotic testing images.

An example of the ten best and worst reconstructed typical and exotic sources produced by the ensemble, selected by their average NCC score, are shown again along with their reconstruction and the residual for each of the three autoencoders in Figures 4.12, 4.13 and 4.14. The first one has the same setup as the previous encoders, so the result is quite similar. The second, however, has a much more constrained latent space and has much more difficulty reconstructing the images as a result. As the goal is not to accurately encode and reconstruct the images this isn't a problem. Ideally it will have more difficulty reconstructing unusual sources than the typical sources it already struggles with. It has more significant residuals, although the largest residuals do appear in the worst reconstructed exotics, as is expected. The final encoder has the largest latent space so is expected to have the best reconstructions and the smallest residuals for typical and exotic. However, again as the aim is not to encode and reconstruct images this does not mean that it is the best of

the three.

Again both the averaging and voting methods were used to classify the sources. Figure 4.15 shows the ROC curves for both the averaging and voting methods for an instance of the ensemble. This time there is a significant difference with the averaging method achieving an AUC of only 0.66 while the voting method manages an AUC of 0.74.

Again, the thresholds for  $F_1$  and  $F_2$  scores, as well as the scores themselves and confusion matrices were averaged over five runs with the split between training data and testing data randomised in each run. Half of the testing data is again used in each run to determine the best threshold and that threshold is then tested with the other half. Figure 4.16 shows the confusion matrices for  $F_1$  and  $F_2$  by the averaging method. The performance here is poorer than that of the ensemble of three similar autoencoders by average. For the  $F_1$  score it averages 0.43 and seems to miss a few more exotic sources while for the  $F_2$  score, an average across the five runs of 0.58, the optimisation has instead included most of the typical sources to optimise the score, making it not very useful for our application. The performance of the ensemble by using the voting method, shown in Figure 4.17, however, yields a better AUC of about 0.74 as opposed to about 0.64, being comparable to the ensemble of three similar autoencoders. Here the  $F_1$  is an average of 0.49 with a standard deviation across the five runs of 0.02. Likewise the  $F_2$  score is also improved and averages 0.66 with a standard deviation across the five runs of 0.02. The  $F_1$  and  $F_2$  thresholds are quite similar and using the  $F_2$  threshold instead of the  $F_1$  threshold only raises the average recall from 0.85 to 0.87 with the standard deviation being 0.02 across the five runs. The low average precision of 0.34 means the user will have to look through multiple sources to find the ones of interest.

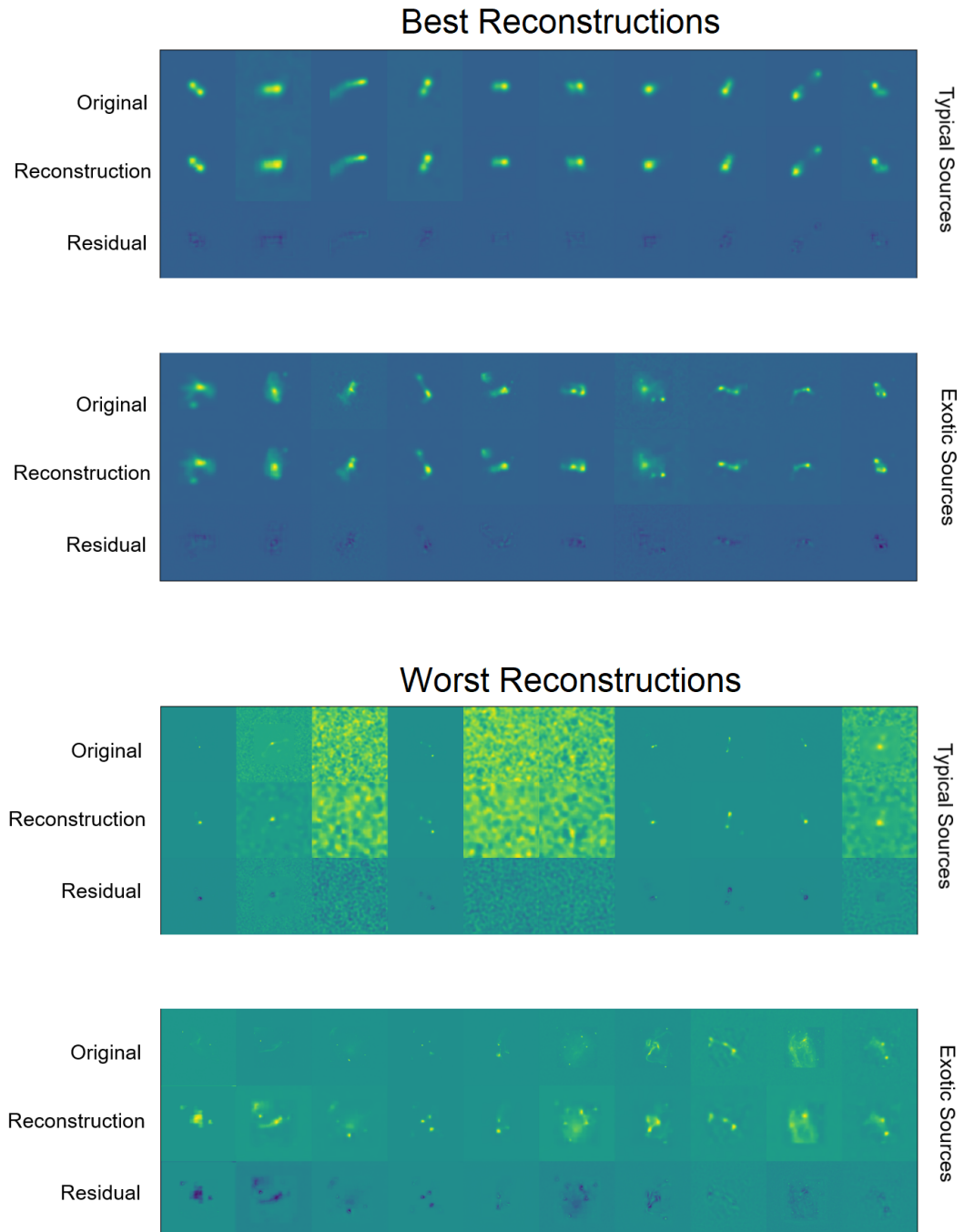


FIGURE 4.12: From the three differing autoencoders the ten best and worst reconstructed sources from both the exotic and typical test sources for the first autoencoder with a latent space of  $16 \times 16 \times 16$ . (Top) The ten best reconstructed exotic and ten best reconstructed typical images from the testing data. For each image the original, the reconstruction and the residual is shown. (Bottom) The ten worst reconstructed exotic and ten worst reconstructed typical images from the testing data.

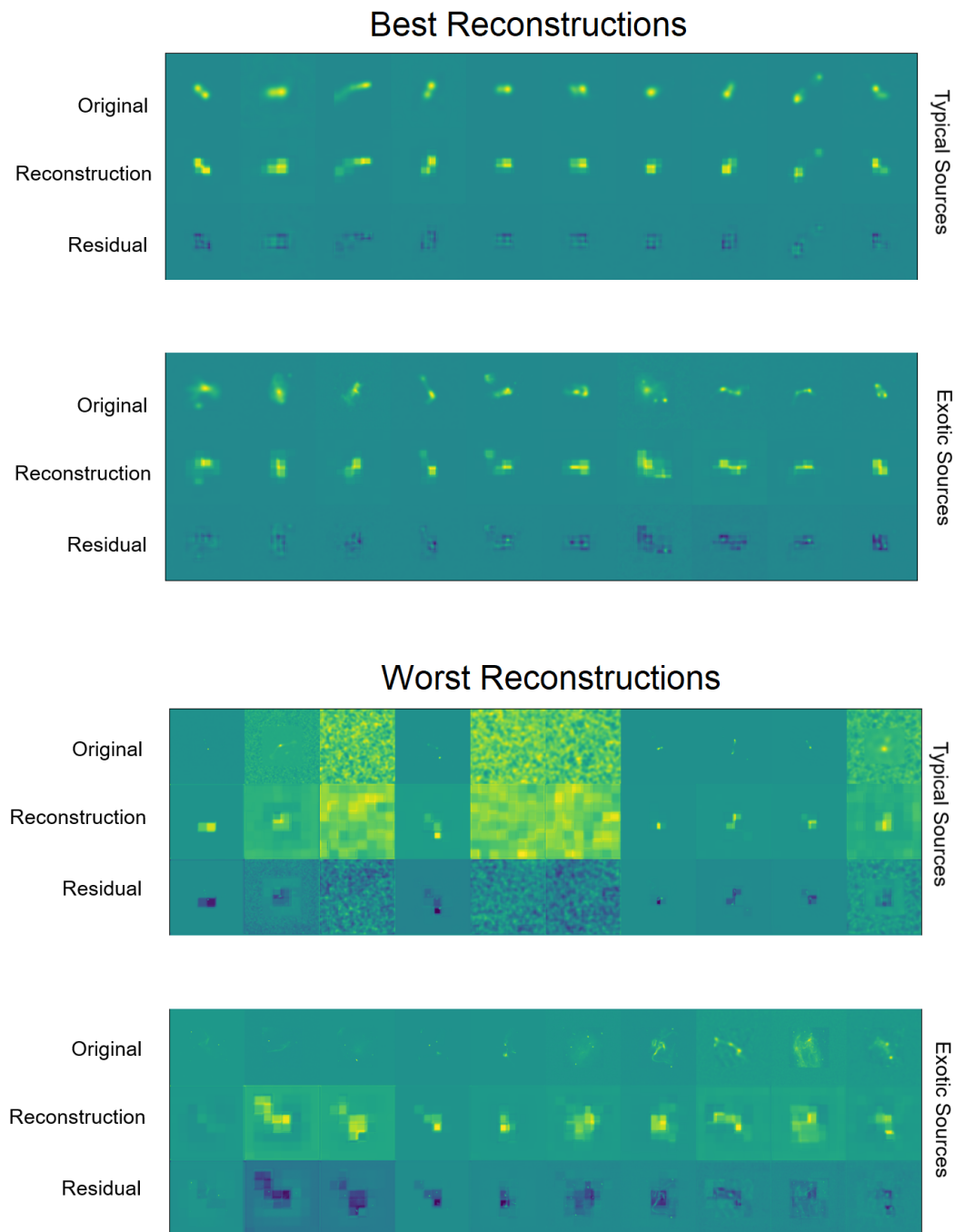


FIGURE 4.13: From the three differing autoencoders, the ten best and worst reconstructed sources from both the exotic and typical test sources, for the second autoencoder with a latent space of  $8 \times 8 \times 32$ . (Top) The ten best reconstructed exotic and ten best reconstructed typical images from the testing data. For each image the original, the reconstruction and the residual is shown. (Bottom) The ten worst reconstructed exotic and ten worst reconstructed typical images from the testing data.

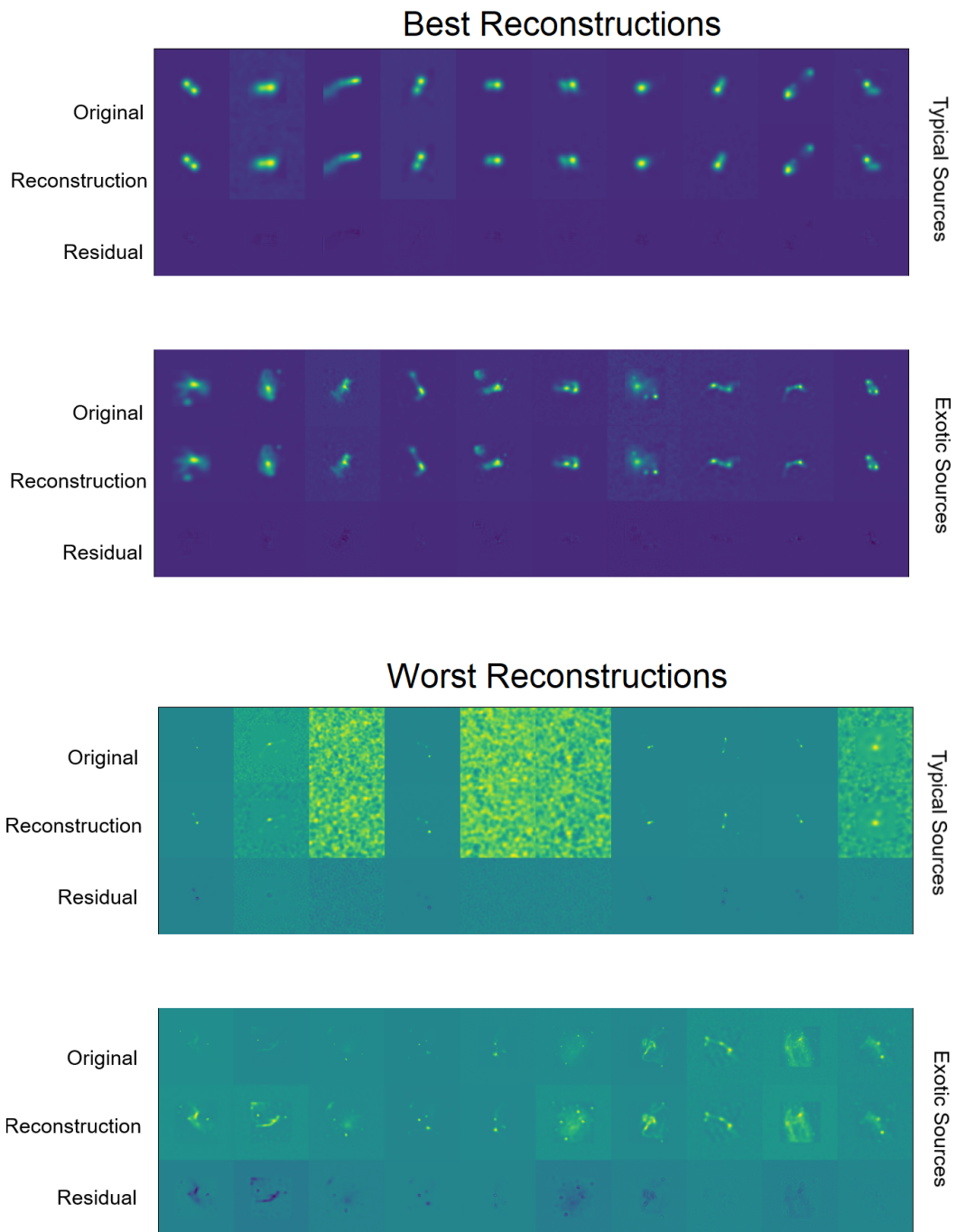


FIGURE 4.14: For the three differing autoencoders, an example of the ten best and worst reconstructed sources from both the exotic and typical test sources it might find. These are from the third autoencoder with a latent space of  $32 \times 32 \times 32$ . (Top) The ten best reconstructed exotic and ten best reconstructed typical images from the testing data. For each image, the original, the reconstruction, and the residual is shown. (Bottom) The ten worst reconstructed exotic and ten worst reconstructed typical images from the testing data.

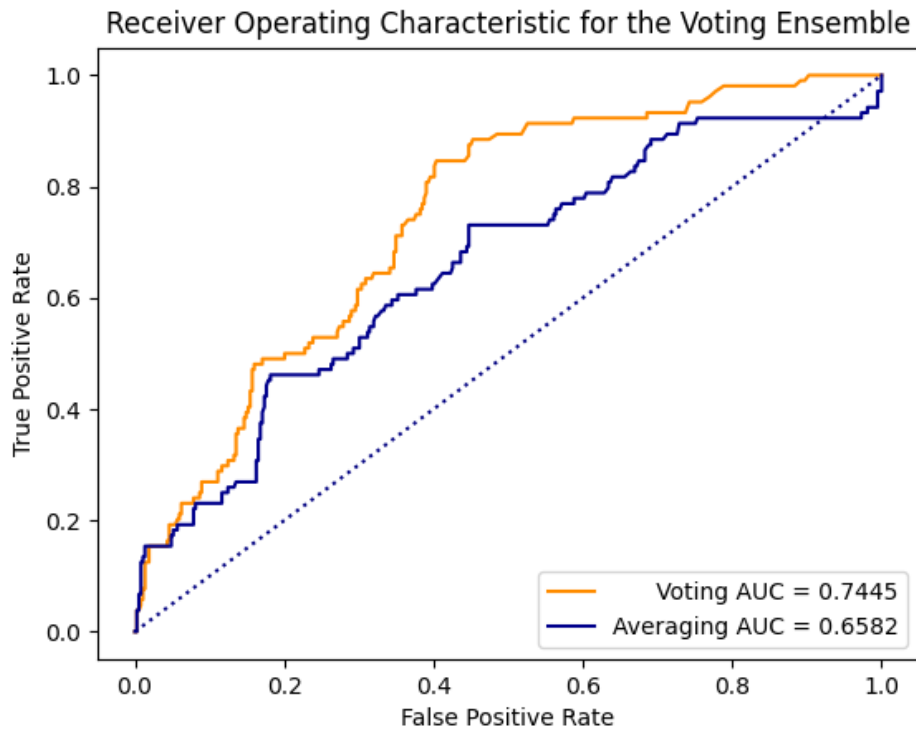


FIGURE 4.15: ROC curves of the three differing autoencoders using the averaging method and the voting methods.

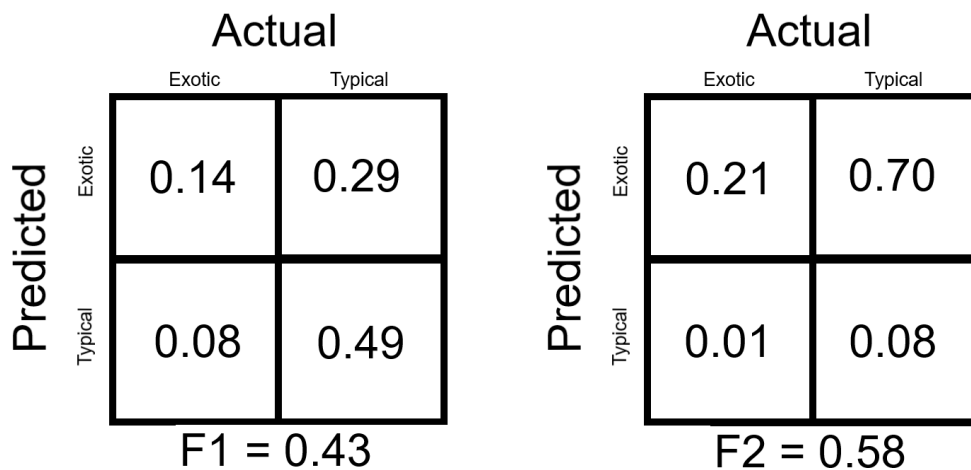


FIGURE 4.16: The  $F_1$  and  $F_2$  scores and confusion matrices for the ensemble of three differing autoencoders using the averaging method. (Left) The confusion matrix and  $F_1$  score of the threshold optimised for  $F_1$ , 0.9254. (Right) The confusion matrix and  $F_2$  score of the threshold optimised for  $F_2$ , 0.9784.



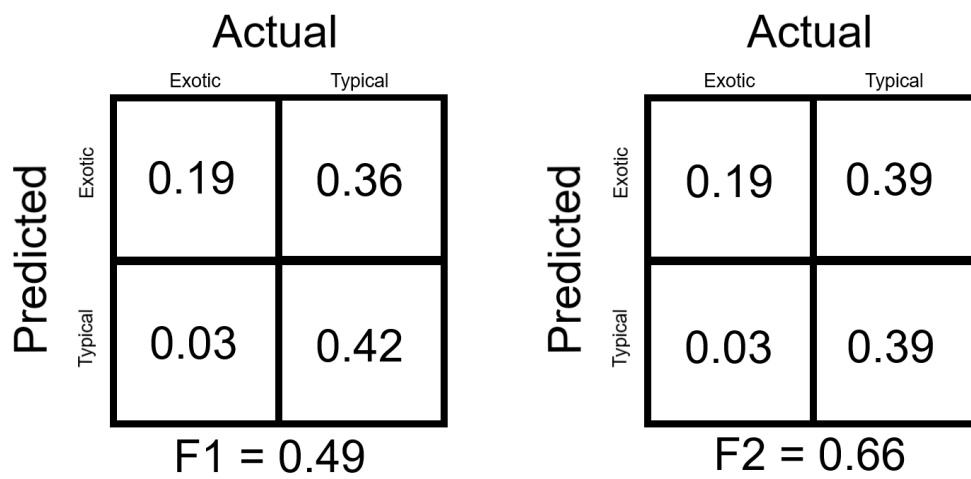


FIGURE 4.17: The  $F_1$  and  $F_2$  scores and confusion matrices for the ensemble of three differing autoencoders using the voting method. (Left) The confusion matrix and  $F_1$  score of the threshold optimised for  $F_1$ , 0.9880. (Right) The confusion matrix and  $F_2$  score of the threshold optimised for  $F_2$ , 0.9892.

## 4.4 Ensemble of Nine Different Autoencoders

Finally, an ensemble of nine autoencoders is tried. It has three of each of the different types of autoencoders described above with latent spaces of  $16 \times 16 \times 16$ ,  $8 \times 8 \times 32$  and  $32 \times 32 \times 32$ . For one run of the ensemble, the histograms of the average NCC reconstruction scores for the typical and exotic testing images are shown in Figure 4.18. Qualitatively, the distributions here look very different from the previous encoders.

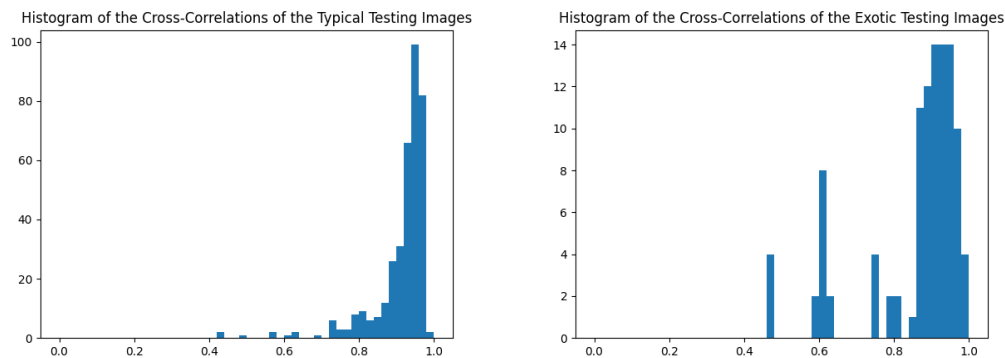


FIGURE 4.18: An example of the reconstruction ability of the ensemble of nine autoencoders. (Left) Histogram of the distribution of the average NCC values of the typical testing images. (Right) Histogram of the average NCC values of the exotic testing images.

Again both the averaging and voting methods are used to sort the images. ROC curves for the two methods for a run of the ensemble are shown in Figure 4.19. Again, the averaging method appears to have a significantly poorer AUC of about 0.65 compared to the voting method at around 0.76.

Finally, the  $F_1$  and  $F_2$  thresholds, scores and confusion matrices are found for the two methods the same way they were for the previous autoencoders by averaging the values over five runs. Again, we see that those taken by averaging perform more poorly than those taken by voting. The averaging method produces an average  $F_1$  score of 0.42 and an average  $F_2$  score of 0.60. The voting method has somewhat improved scores with an average  $F_1$  score of 0.49 with standard deviation 0.03 across the five runs and an average  $F_2$  score of 0.64 with a standard deviation of 0.04 across the five runs. At the  $F_2$  threshold the average recall is high at 0.89, however the precision is just 0.31 so the user will need to look through multiple sources to find those of interest. The  $F_1$  and  $F_2$  scores are similar to those of the best of the previous autoencoders as are their standard deviations. Similarly, the majority of exotic sources seem to be correctly classified while the typical sources are closer to being split in half.

So, the output can be used to more easily search for exotic sources by greatly reducing the number of typical sources while managing to keep most of the exotic sources.

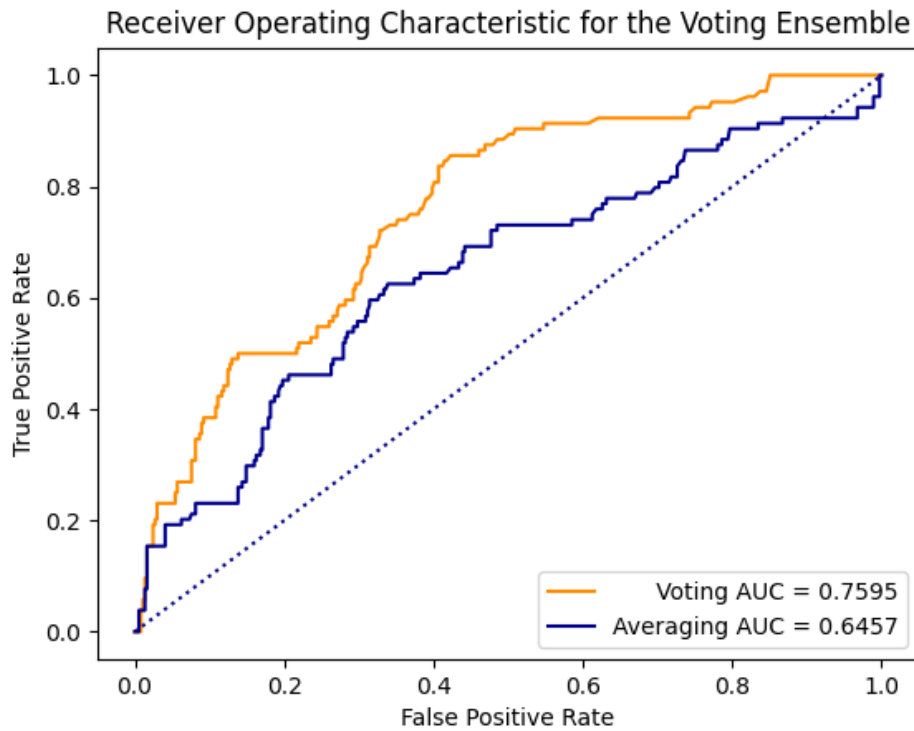


FIGURE 4.19: The ROC curves for a run of the ensemble of nine autoencoders showing both the averaging method and voting method.

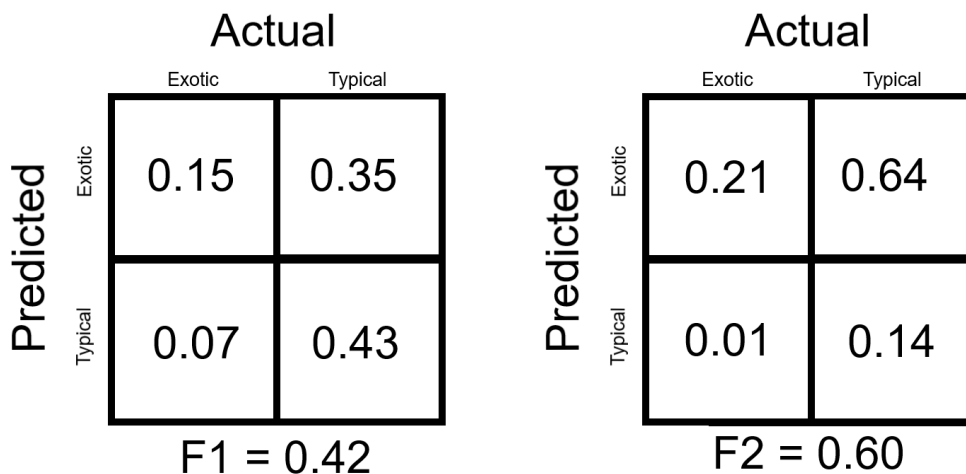


FIGURE 4.20: The  $F_1$  and  $F_2$  scores and confusion matrices for the ensemble of nine autoencoders using the averaging method. (Left) The confusion matrix and  $F_1$  score of the threshold optimised for  $F_1$ , 0.9339. (Right) The confusion matrix and  $F_2$  score of the threshold optimised for  $F_2$ , 0.9699.

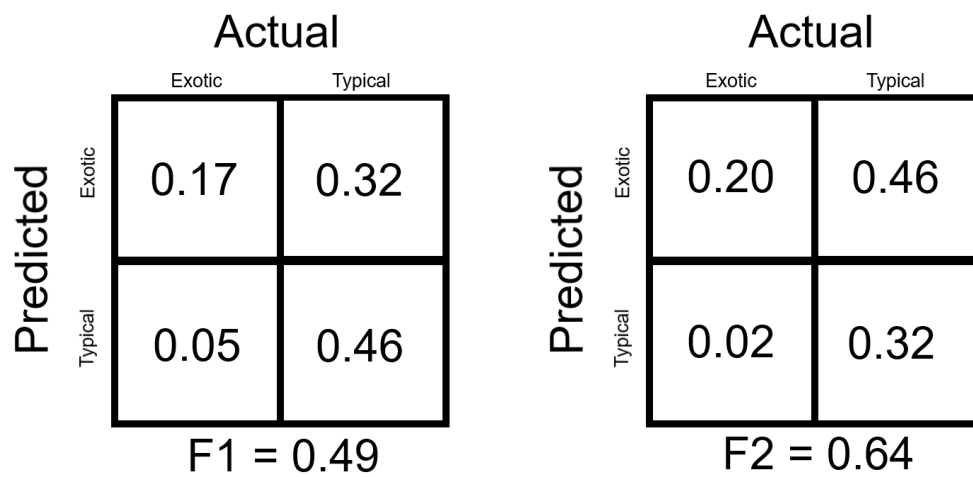


FIGURE 4.21: The  $F_1$  and  $F_2$  scores and confusion matrices for the ensemble of nine autoencoders using the voting method. (Left) The confusion matrix and  $F_1$  score of the threshold optimised for  $F_1$ , 0.9780. (Right) The confusion matrix and  $F_2$  score of the threshold optimised for  $F_2$ , 0.9884.

## 4.5 A Sample of Automatically Selected Interesting Sources

To demonstrate the sources that may be found in this data in this manner, Figure 4.22 contains a sample of interesting and exotic sources automatically identified. The sources classified as exotic were inspected and a sample of the most interesting sources was manually selected. The sources also demonstrate the quality of the MG-CLS data. The sources are good candidates for further study including looking at multiwavelength data, which is well beyond the scope of this thesis.

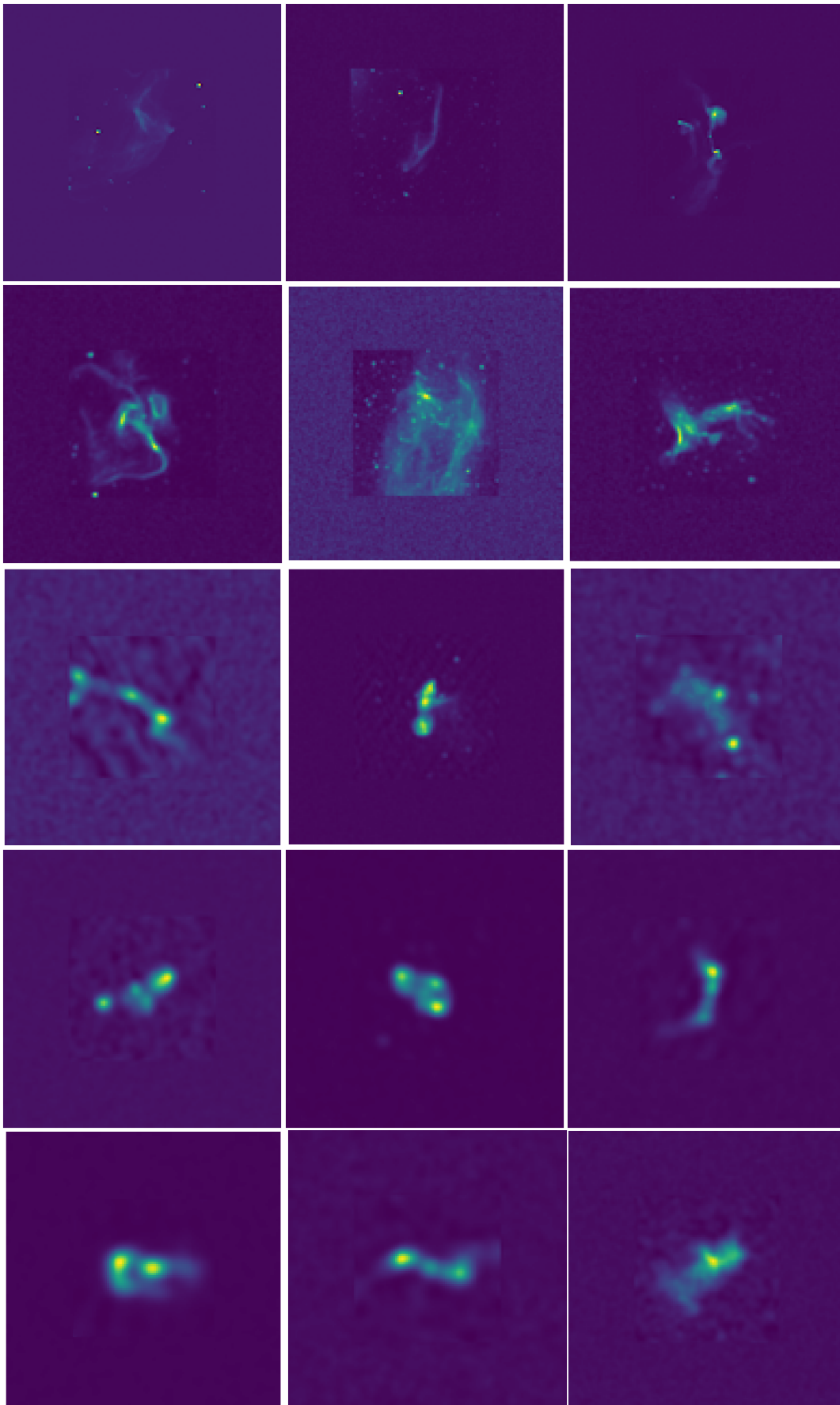


FIGURE 4.22: A manual selection of the testing images marked as exotic by the autoencoders.

## Chapter 5

# Conclusions

When assessing the results it is clear that the autoencoders show some promise in finding exotic sources. The classifications returned by the algorithms are not, however, a reliable classification that may be relied upon without inspection. For well selected thresholds, the algorithms return most of the exotic sources. However, they also return a fairly high number of false positives. This is not dissimilar to what was seen with other approaches such as `Astronomy`, where recall was also prioritised. The use case and value in these approaches, then is to help the user search through large collections of potential sources by reducing the total number without discarding the exotic sources. The algorithms could potentially remove half or more of the typical sources while retaining almost all of the exotic sources. As a result the  $F_1$  performance was always poor, as they have difficulty with precision, but the  $F_2$  performance that prioritises recall is much more promising. Although they do not produce reliable classifications that may be used without further analysis, the models succeed at least partially in the aiding in isolating exotic radio sources by significantly reducing the time required to search for such sources. Autoencoders certainly show potential in finding exotic radio sources by morphology in survey data as per the objective of this thesis.

When it comes to the ensembles of autoencoders, it is clear that the voting method is superior to the averaging method. Although numerous arrangements of ensembles were tried, their performance when using a voting approach remained mostly similar. This may indicate that the certain exotic sources are morphologically too similar to many typical sources for this type of approach. The standard deviation across the runs did not improve with ensembling either. This may be due to the ensembling method used, as opposed to a more typical implementation of bagging or a more complex means of ensembling such as stacking. Many sources may be classified as exotic by experts because of a deeper understanding of the source itself while not having features that are particularly unusual in shape. So, for automated classification a method that does not purely rely on morphology might also be worth considering. It should also be noted that the categories used here of "typical" and "exotic" are very broad. Searching more specifically for more narrowly defined type of morphology, such as X-shaped sources or bent tail sources, may yield better results. The results could also change if a larger set of data is used, as the dataset used here is relatively small at just 3700 sources.

With the ever increasing power of modern radio telescopes, it has become infeasible to manually search through or classify the hundreds of thousands or even millions of sources in modern and future surveys. Therefore, the ability to search through this data in an automated fashion has become extremely important. The power of these telescopes means that many of these sources are now visible at much

greater angular resolution and sensitivity, sometimes making them difficult to even classify manually, and often leaving their fields crowded with background point sources and other sources. For this reason, even more powerful automated tools will become required to handle these more complicated challenges for automated classification. Autoencoders prove themselves a promising candidate for further development in this area. More testing is certainly required with larger catalogues of sources, as with more variations of the algorithms and possibly looking for more specific types of sources. The results here did not indicate a level of accuracy suitable for a catalogue, however, when trying to search the data for certain types of sources these automated approaches are already capable of returning most suitable candidates and rejecting many unsuitable ones for even a search as broad as "exotic" morphologies. In a search for some more narrowly defined morphology it may do even better, a topic for future work as well as the more detailed multi-wavelength analysis of individual sources of interest identified so far.

For further future work it would be very useful to further investigate the latent space of autoencoders and determine if the classes can separate here in the manner more typical of autoencoders. It would also be useful to further study the ensembling of the autoencoders and determine if the performance of the ensembles may be improved with a different ensembling technique, such as bagging, or if it can be determined if ensembling does not work well for this particular type of data. It would also be useful to compare these particular models against a wider variety of machine learning approaches to classification on the same data in order to determine which approaches show the most promise on these types of datasets.



# Bibliography

- Abazajian, Kevork N. et al. (June 2009). "The Seventh Data Release of the Sloan Digital Sky Survey". In: *Astrophysical Journal, Supplement* 182.2, pp. 543–558. DOI: [10.1088/0067-0049/182/2/543](https://doi.org/10.1088/0067-0049/182/2/543).
- Aniyan, A. K. and K. Thorat (June 2017). "Classifying Radio Galaxies with the Convolutional Neural Network". In: *Astrophysical Journal, Supplement* 230.2, 20, p. 20. DOI: [10.3847/1538-4365/aa7333](https://doi.org/10.3847/1538-4365/aa7333).
- Astropy Collaboration et al. (Oct. 2013). "Astropy: A community Python package for astronomy". In: *Astronomy and Astrophysics* 558, A33, A33. DOI: [10.1051/0004-6361/201322068](https://doi.org/10.1051/0004-6361/201322068).
- Astropy Collaboration et al. (Sept. 2018). "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package". In: *Astronomical Journal* 156.3, 123, p. 123. DOI: [10.3847/1538-3881/aabc4f](https://doi.org/10.3847/1538-3881/aabc4f).
- Banfield, J. K. et al. (Nov. 2015). "Radio Galaxy Zoo: host galaxies and radio morphologies derived from visual inspection". In: *Monthly Notices of the Royal Astronomical Society* 453.3, pp. 2326–2340. DOI: [10.1093/mnras/stv1688](https://doi.org/10.1093/mnras/stv1688).
- Begelman, M. C., R. D. Blandford, and M. J. Rees (1980). "Massive black hole binaries in active galactic nuclei". In: *Nature* 287, pp. 307–309. DOI: [10.1038/287307a0](https://doi.org/10.1038/287307a0).
- Bianchi, S., R. Maiolino, and G. Risaliti (2012). "AGN Obscuration and the Unified Model". In: *Advances in Astronomy* vol. 2012, pp. 1–17. DOI: [10.1155/2012/782030](https://doi.org/10.1155/2012/782030).
- Blandford, R., D. Meier, and A. Readhead (2019). "Relativistic Jets from Active Galactic Nuclei". In: *Annual Review of Astronomy and Astrophysics* 57.1, pp. 467–509. DOI: [10.1146/annurev-astro-081817-051948](https://doi.org/10.1146/annurev-astro-081817-051948).
- Bolton, J. G., A. Savage, and A. E. Wright (Jan. 1979). "The Parkes 2700 MHz survey (14th part) : catalogue and new optical identifications." In: *Australian Journal of Physics Astrophysical Supplement* 46, pp. 1–27.
- Boureau, Y. Lan, Jean Ponce, and Yann Lecun (2010). "A theoretical analysis of feature pooling in visual recognition". English (US). In: *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*. ICML 2010 - Proceedings, 27th International Conference on Machine Learning, ICML 2010 ; Conference date: 21-06-2010 Through 25-06-2010, pp. 111–118. ISBN: 9781605589077.
- Bourlard, Herve and Y Kamp (Feb. 1988). "Auto-Association by Multilayer Perceptrons and Singular Value Decomposition". In: *Biological cybernetics* 59, pp. 291–4. DOI: [10.1007/BF00332918](https://doi.org/10.1007/BF00332918).
- Breiman, L. (1994). "Bagging predictors". In: *Machine Learning* 24, pp. 123–140. DOI: <https://doi.org/10.1007/BF00058655>.
- Breunig, Markus M. et al. (2000). "LOF: Identifying Density-Based Local Outliers". In: *SIGMOD Rec.* 29.2, 93–104. ISSN: 0163-5808. DOI: [10.1145/335191.335388](https://doi.org/10.1145/335191.335388). URL: <https://doi.org/10.1145/335191.335388>.
- Bridle, Alan (2006). *Images of Radio Galaxies and Quasars*. URL: <https://www.cv.nrao.edu/~abridle/images.htm>. (accessed: December 2019).

- Chandola, Varun, Arindam Banerjee, and Vipin Kumar (2009). "Anomaly Detection: A Survey". In: *ACM Comput. Surv.* 41.3. ISSN: 0360-0300. DOI: [10.1145/1541880.1541882](https://doi.org/10.1145/1541880.1541882). URL: <https://doi.org/10.1145/1541880.1541882>.
- Chollet, François et al. (2015). *Keras*. <https://keras.io>.
- Condon, J. J. (Dec. 1984a). "Cosmological evolution of radio sources." In: *Astrophysical Journal* 287, pp. 461–474. DOI: [10.1086/162705](https://doi.org/10.1086/162705).
- (Sept. 1984b). "Cosmological evolution of radio sources found at 1.4 GHz". In: *Astrophysical Journal* 284, pp. 44–53. DOI: [10.1086/162382](https://doi.org/10.1086/162382).
- Condon, J. J. et al. (May 1998). "The NRAO VLA Sky Survey". In: *Astronomical Journal* 115.5, pp. 1693–1716. DOI: [10.1086/300337](https://doi.org/10.1086/300337).
- Condon, J. J. et al. (2012). "Resolving the Radio Source Background: Deeper Understanding through Confusion". In: *The Astrophysical Journal* 758.1. DOI: [10.1088/0004-637X/758/1/23](https://doi.org/10.1088/0004-637X/758/1/23).
- Cotton, W. D. et al. (June 2020). "Hydrodynamical backflow in X-shaped radio galaxy PKS 2014-55". In: *Monthly Notices of the Royal Astronomy Society* 495.1, pp. 1271–1283. DOI: [10.1093/mnras/staa1240](https://doi.org/10.1093/mnras/staa1240).
- Croston, J. H. et al. (Apr. 2005). "Jet/Environment Interactions of FRI and FR II Radio Galaxies". In: *X-Ray and Radio Connections*. Ed. by L. O. Sjouwerman and K. K. Dyer, 7.06, p. 7.06. arXiv: [astro-ph/0404440](https://arxiv.org/abs/astro-ph/0404440) [[astro-ph](https://arxiv.org/abs/astro-ph)].
- Ding, Kaize et al. (2019). "Deep Anomaly Detection on Attributed Networks". In: *SDM*.
- Doorenbos, Lars et al. (Apr. 2021). "Comparison of Outlier Detection Methods on Astronomical Image Data". In: pp. 197–223. ISBN: 978-3-030-65866-3. DOI: [10.1007/978-3-030-65867-0\\_9](https://doi.org/10.1007/978-3-030-65867-0_9).
- Eilek, Jean A. (2014). *Active Galaxies and Quasistellar Objects, Jets*. URL: <https://ned.ipac.caltech.edu/level5/ESSAYS/Eilek/eilek.html>. (accessed: December 2019).
- Fanaroff, B. L. and J. M. Riley (1974). "The Morphology of Extragalactic Radio Sources of High and Low Luminosity". In: *Monthly Notices of the Royal Astronomical Society* 167.1, 31P–36P. DOI: [10.1093/mnras/167.1.31p](https://doi.org/10.1093/mnras/167.1.31p).
- Fawcett, Tom (2006). "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8. ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2005.10.010>. URL: <https://www.sciencedirect.com/science/article/pii/S016786550500303X>.
- Ferrari, C. et al. (Feb. 2008). "Observations of Extended Radio Emission in Clusters". In: *Space Science Reviews* 134.1-4, pp. 93–118. DOI: [10.1007/s11214-008-9311-x](https://doi.org/10.1007/s11214-008-9311-x).
- Fraknoi, Andrew et al. (2016). *Astronomy*. Houston, USA: OpenStax.
- Freund, Yoav and Robert E Schapire (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1, pp. 119–139. ISSN: 0022-0000. DOI: <https://doi.org/10.1006/jcss.1997.1504>. URL: <https://www.sciencedirect.com/science/article/pii/S002200009791504X>.
- Goedhart, Sharmila, Vasaant Krishnan, and Fernando Camilo (2022). *Meerkat specifications*. URL: <https://skaafrica.atlassian.net/wiki/spaces/ESDKB/pages/277315585/MeerKAT+specifications>.
- Gong, Dong et al. (Oct. 2019). "Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection". In: pp. 1705–1714. DOI: [10.1109/ICCV.2019.00179](https://doi.org/10.1109/ICCV.2019.00179).
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press.

- Gopal-Krishna et al. (Feb. 2012). "On the origin of X-shaped radio galaxies". In: *Research in Astronomy and Astrophysics* 12.2, pp. 127–146. DOI: [10.1088/1674-4527/12/2/002](https://doi.org/10.1088/1674-4527/12/2/002).
- Grauman, K. and T. Darrell (2005). "The pyramid match kernel: discriminative classification with sets of image features". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2, 1458–1465 Vol. 2. DOI: [10.1109/ICCV.2005.239](https://doi.org/10.1109/ICCV.2005.239).
- Hajer, Jan et al. (Apr. 2020). "Novelty detection meets collider physics". In: *Physical Review D* 101. DOI: [10.1103/PhysRevD.101.076015](https://doi.org/10.1103/PhysRevD.101.076015).
- Hanley, J A and B J McNeil (1982). "The meaning and use of the area under a receiver operating characteristic (ROC) curve." In: *Radiology* 143.1. PMID: 7063747, pp. 29–36. DOI: [10.1148/radiology.143.1.7063747](https://doi.org/10.1148/radiology.143.1.7063747). eprint: <https://doi.org/10.1148/radiology.143.1.7063747>. URL: <https://doi.org/10.1148/radiology.143.1.7063747>.
- Hasan, Mahmudul et al. (June 2016). "Learning Temporal Regularity in Video Sequences". In: pp. 733–742. DOI: [10.1109/CVPR.2016.86](https://doi.org/10.1109/CVPR.2016.86).
- He, Kaiming et al. (2014). "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, pp. 346–361. ISBN: 978-3-319-10578-9. DOI: [10.1007/978-3-319-10578-9\\_23](https://doi.org/10.1007/978-3-319-10578-9_23).
- Hodge, Victoria and Jim Austin (2004). "A Survey of Outlier Detection Methodologies". In: *Artif. Intell. Rev.* 22.2, 85–126. ISSN: 0269-2821. DOI: [10.1023/B:AIRE.0000045502.10941.a9](https://doi.org/10.1023/B:AIRE.0000045502.10941.a9). URL: <https://doi.org/10.1023/B:AIRE.0000045502.10941.a9>.
- Hotelling, H. (1933). "Analysis of a complex of statistical variables into principal components". In: *Journal of Educational Psychology* 24, pp. 417–441. DOI: [10.1037/h0071325](https://doi.org/10.1037/h0071325).
- Knowles, K. et al. (2022). "The MeerKAT Galaxy Cluster Legacy Survey". In: *Astronomy Astrophysics* 657, A56. ISSN: 1432-0746. DOI: [10.1051/0004-6361/202141488](https://doi.org/10.1051/0004-6361/202141488). URL: <http://dx.doi.org/10.1051/0004-6361/202141488>.
- Kohonen, Teuvo (1988). *Self-Organization and Associative Memory*.  
— (2001). *Self-Organizing Maps*.
- Kormendy, J. and D. Richstone (1995). "Inward Bound—The Search for Supermassive Black Holes in Galactic Nuclei". In: *Annual Review of Astronomy and Astrophysics* 33.1, pp. 581–624. DOI: [10.1146/annurev.aa.33.090195.003053](https://doi.org/10.1146/annurev.aa.33.090195.003053).
- Kwon, Donghwoon et al. (2017). "A survey of deep learning-based network anomaly detection". In: *Cluster Computing* 22, pp. 949–961.
- Lazebnik, S., C. Schmid, and J. Ponce (2006). "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2, pp. 2169–2178. DOI: [10.1109/CVPR.2006.68](https://doi.org/10.1109/CVPR.2006.68).
- Leahy, J. P. and A. G. Williams (Oct. 1984). "The bridges of classical double radio sources." In: *Monthly Notices of the Royal Astronomy Society* 210, pp. 929–951. DOI: [10.1093/mnras/210.4.929](https://doi.org/10.1093/mnras/210.4.929).
- LeCun, Y. et al. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4, 541–551. DOI: [doi:10.1162/neco.1989.1.4.541](https://doi.org/10.1162/neco.1989.1.4.541).
- LeCun, Yann (June 1987). *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*. English (US). Universite P. et M. Curie (Paris 6).

- Lee, Chen-Yu, Patrick W. Gallagher, and Zhuowen Tu (2016). "Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Vol. 51. Proceedings of Machine Learning Research. Cadiz, Spain: PMLR, pp. 464–472. URL: <https://proceedings.mlr.press/v51/lee16a.html>.
- Lintott, Chris et al. (Dec. 2010). "Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies\*". In: *Monthly Notices of the Royal Astronomical Society* 410.1, pp. 166–178. ISSN: 0035-8711. DOI: [10.1111/j.1365-2966.2010.17432.x](https://doi.org/10.1111/j.1365-2966.2010.17432.x). eprint: <https://academic.oup.com/mnras/article-pdf/410/1/166/18442057/mnras0410-0166.pdf>. URL: <https://doi.org/10.1111/j.1365-2966.2010.17432.x>.
- Lintott, Chris J. et al. (Sept. 2008). "Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey\*". In: *Monthly Notices of the Royal Astronomical Society* 389.3, pp. 1179–1189. ISSN: 0035-8711. DOI: [10.1111/j.1365-2966.2008.13689.x](https://doi.org/10.1111/j.1365-2966.2008.13689.x). eprint: <https://academic.oup.com/mnras/article-pdf/389/3/1179/3325962/mnras0389-1179.pdf>. URL: <https://doi.org/10.1111/j.1365-2966.2008.13689.x>.
- Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou (2008). "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. DOI: [10.1109/ICDM.2008.17](https://doi.org/10.1109/ICDM.2008.17).
- Lloyd, S. (1982). "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2, pp. 129–137. DOI: [10.1109/TIT.1982.1056489](https://doi.org/10.1109/TIT.1982.1056489).
- Lochner, M. and B.A. Bassett (2021). "Astronomy: Personalised active anomaly detection in astronomical data". In: *Astronomy and Computing* 36, p. 100481. ISSN: 2213-1337. DOI: [10.1016/j.ascom.2021.100481](https://doi.org/10.1016/j.ascom.2021.100481). URL: <http://dx.doi.org/10.1016/j.ascom.2021.100481>.
- Lu, Weining et al. (Sept. 2017). "Unsupervised Sequential Outlier Detection with Deep Architectures". English (US). In: *IEEE Transactions on Image Processing* 26.9. Publisher Copyright: © 1992-2012 IEEE., pp. 4321–4330. ISSN: 1057-7149. DOI: [10.1109/TIP.2017.2713048](https://doi.org/10.1109/TIP.2017.2713048).
- McCulloch, Warren S. and Pitts Walter (Dec. 1943). "A logical calculus of the ideas immanent in nervous activity". In: *The Bulletin of Mathematical Biophysics* 5.4, pp. 115–133. ISSN: 0007-4985. DOI: [10.1007/bf02478259](https://doi.org/10.1007/bf02478259). URL: <https://cir.nii.ac.jp/crid/1360574095344592000>.
- Mingo, B. et al. (Apr. 2022). "Accretion mode versus radio morphology in the LOFAR Deep Fields". In: *Monthly Notices of the Royal Astronomy Society* 511.3, pp. 3250–3271. DOI: [10.1093/mnras/stac140](https://doi.org/10.1093/mnras/stac140).
- Mitchell, K. J. and J. J. Condon (Oct. 1985). "A confusion-limited 1.49-GHz VLA survey centered on alpha=13h00m 37s, delta = +30 34." In: *Astronomical Journal* 90, pp. 1957–1966. DOI: [10.1086/113899](https://doi.org/10.1086/113899).
- Mohan, Niruj and David Rafferty (Feb. 2015). *PyBDSF: Python Blob Detection and Source Finder*. ascl: [1502.007](https://ascl.net/1502.007).
- Morganti, R. et al. (Jan. 1988). "Low luminosity radio galaxies : effects of gaseous environment." In: *Astronomy and Astrophysics* 189, pp. 11–26.
- Mostert, Rafaël I. J. et al. (Jan. 2021). "Unveiling the rarest morphologies of the LOFAR Two-metre Sky Survey radio source population with self-organised maps". In: *Astronomy and Astrophysics* 645, A89, A89. DOI: [10.1051/0004-6361/202038500](https://doi.org/10.1051/0004-6361/202038500).
- Nassif, Ali Bou et al. (2021). "Machine Learning for Anomaly Detection: A Systematic Review". In: *IEEE Access* 9, pp. 78658–78700. DOI: [10.1109/ACCESS.2021.3083060](https://doi.org/10.1109/ACCESS.2021.3083060).

- Nielsen, Michael A. (2015). In: *Neural Networks and Deep Learning*. Determination Press.
- Owen, F. N. and M. J. Ledlow (Jan. 1994). "The FRI/II Break and the Bivariate Luminosity Function in Abell Clusters of Galaxies". In: *The Physics of Active Galaxies*. Ed. by Geoffrey V. Bicknell, Michael A. Dopita, and Peter J. Quinn. Vol. 54. Astronomical Society of the Pacific Conference Series, p. 319.
- Owen, Frazer N. and Robert A. Laing (May 1989). "CCD surface photometry of radio galaxies – I. FR class I and II sources". In: *Monthly Notices of the Royal Astronomical Society* 238.2, pp. 357–378. ISSN: 0035-8711. DOI: [10.1093/mnras/238.2.357](https://doi.org/10.1093/mnras/238.2.357). eprint: <https://academic.oup.com/mnras/article-pdf/238/2/357/3885076/mnras238-0357.pdf>. URL: <https://doi.org/10.1093/mnras/238.2.357>.
- Owen, Frazer N. and G. E. Morrison (2008). "THE DEEP SWIRE FIELD. I. 20 cm CONTINUUM RADIO OBSERVATIONS: A CROWDED SKY". In: *The Astronomical Journal* 136.5, pp. 1889–1900. DOI: [10.1088/0004-6256/136/5/1889](https://doi.org/10.1088/0004-6256/136/5/1889). URL: <https://doi.org/10.1088/0004-6256/136/5/1889>.
- Pala, Tuba et al. (Sept. 2018). "Comparison of Pooling Methods for Handwritten Digit Recognition Problem". In: pp. 1–5. DOI: [10.1109/IDAP.2018.8620848](https://doi.org/10.1109/IDAP.2018.8620848).
- Pearson, Karl (1901). "LIII. On lines and planes of closest fit to systems of points in space". In: *Philosophical Magazine Series 1* 2, pp. 559–572.
- Polsterer, K. L., F. Gieseke, and C. Igel (Sept. 2015). "Automatic Galaxy Classification via Machine Learning Techniques: Parallelized Rotation/Flipping INvariant Kohonen Maps (PINK)". In: *Astronomical Data Analysis Software and Systems XXIV (ADASS XXIV)*. Ed. by A. R. Taylor and E. Rosolowsky. Vol. 495. Astronomical Society of the Pacific Conference Series, p. 81.
- Roberts, Ethan, Bruce A. Bassett, and Michelle Lochner (2020). *Bayesian Anomaly Detection and Classification for Noisy Data*. DOI: [10.3233/HIS-200282](https://doi.org/10.3233/HIS-200282).
- Rosenblatt, Frank (1957). *The Perceptron - A Perceiving and Recognizing Automaton*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY.
- (1961). *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY.
- Rottmann, H. (Aug. 2001). "Jet-Reorientation in X-shaped Radio Galaxies". PhD thesis. -.
- Rudnick, L. (Apr. 2002). "Simple Multiresolution Filtering and the Spectra of Radio Galaxies and Supernova Remnants". In: *Publications of the ASP* 114.794, pp. 427–449. DOI: [10.1086/342499](https://doi.org/10.1086/342499).
- Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (1986). "Learning representations by back-propagating errors". In: *Nature* 323.6088, 533–536. DOI: <https://doi.org/10.1038/323533a0>.
- Shimwell, T. W. et al. (Feb. 2017). "The LOFAR Two-metre Sky Survey. I. Survey description and preliminary data release". In: *Astronomy and Astrophysics* 598, A104, A104. DOI: [10.1051/0004-6361/201629313](https://doi.org/10.1051/0004-6361/201629313).
- Swart, Gerhard P., Peter E. Dewdney, and Andrea Cremonini (2022). "Highlights of the SKA1-Mid telescope architecture". In: *Journal of Astronomical Telescopes, Instruments, and Systems* 8.1, pp. 1–23. DOI: [10.1117/1.JATIS.8.1.011021](https://doi.org/10.1117/1.JATIS.8.1.011021). URL: <https://doi.org/10.1117/1.JATIS.8.1.011021>.
- Tang, H. et al. (Mar. 2022). "Radio Galaxy Zoo: giant radio galaxy classification using multidomain deep learning". In: *Monthly Notices of the Royal Astronomy Society* 510.3, pp. 4504–4524. DOI: [10.1093/mnras/stab3553](https://doi.org/10.1093/mnras/stab3553).
- Tharwat, Alaa (2020). "Classification assessment methods". In: *Applied Computing and Informatics*.

- Töscher, Andreas and Michael Jahrer (Jan. 2009). "The BigChaos Solution to the Netflix Grand Prize". In.
- van Haarlem, M. P. et al. (Aug. 2013). "LOFAR: The LOw-Frequency ARray". In: *Astronomy and Astrophysics* 556, A2, A2. DOI: [10.1051/0004-6361/201220873](https://doi.org/10.1051/0004-6361/201220873).
- Werbos, Paul (Jan. 1974). "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Thesis (Ph. D.). Appl. Math. Harvard University". PhD thesis.
- Willett, Kyle W. et al. (Sept. 2013). "Galaxy Zoo 2: detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey". In: *Monthly Notices of the Royal Astronomical Society* 435.4, pp. 2835–2860. ISSN: 0035-8711. DOI: [10.1093/mnras/stt1458](https://doi.org/10.1093/mnras/stt1458). eprint: <https://academic.oup.com/mnras/article-pdf/435/4/2835/3372631/stt1458.pdf>. URL: <https://doi.org/10.1093/mnras/stt1458>.
- Wolpert, David H. (1992). "Stacked generalization". In: *Neural Networks* 5.2, pp. 241–259. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1). URL: <https://www.sciencedirect.com/science/article/pii/S0893608005800231>.
- Wright, A. and R. Otrupcek (Jan. 1990). "Parkes Catalog, 1990, Australia telescope national facility." In: *PKS Catalog (1990)*, p. 0.
- Xu, Dan et al. (2015). "Learning Deep Representations of Appearance and Motion for Anomalous Event Detection". In: *BMVC*.
- York, Donald G. et al. (Sept. 2000). "The Sloan Digital Sky Survey: Technical summary". English (US). In: *Astronomical Journal* 120.3, pp. 1579–1587. ISSN: 0004-6256. DOI: [10.1086/301513](https://doi.org/10.1086/301513).
- Zhao, Yiru et al. (2017). "Spatio-Temporal AutoEncoder for Video Anomaly Detection". In: *Proceedings of the 25th ACM International Conference on Multimedia. MM '17*. Mountain View, California, USA: Association for Computing Machinery, 1933–1941. ISBN: 9781450349062. DOI: [10.1145/3123266.3123451](https://doi.org/10.1145/3123266.3123451). URL: <https://doi.org/10.1145/3123266.3123451>.
- Zhou and Chellappa (1988). "Computation of optical flow using a neural network". In: *IEEE 1988 International Conference on Neural Networks*, 71–78 vol.2. DOI: [10.1109/ICNN.1988.23914](https://doi.org/10.1109/ICNN.1988.23914).
- Zirbel, E. L. and S. A. Baum (Jan. 1994). "Are FRI's and FR II's Different?" In: *The Physics of Active Galaxies*. Ed. by Geoffrey V. Bicknell, Michael A. Dopita, and Peter J. Quinn. Vol. 54. Astronomical Society of the Pacific Conference Series, p. 379.
- Zong, Bo et al. (2018). "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection". In: *ICLR*.