DOCTORAL THESIS

Giant Radio Galaxies as Probes of the Low Density Intergalactic Medium

Author:
HONGMING TANG

Supervisor:
Prof. ANNA M.M. SCAIFE

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy

in the

Department of Physics and Astronomy in the School of Natural Sciences
Faculty of Science and Engineering

2021

Contents

Word count: 71770

List of Figures

List of Tables

List of Abbreviations

1C	First (1st) Cambridge catalogue of Radio Sources
2C	Second (2nd) Cambridge catalogue of Radio Sources
3C	Third (3rd) Cambridge catalogue of Radio Sources
3CR	Revised 3C catalogue
3CRR	A Revised sample of bright sources in the 3CR catalogue
AdaGrad	Adaptive Gradient algorithm
Adam	Adaptive moment Estimation method
AGN	Active Galactic Nucleus
ANN	Artificial Neural Network
ASKAP	Australian Square Kilometre Array Pathfinder
ATLAS	Australia Telescope Large Area Survey
AUC	Area Under ROC Curve
BCG	Brightest Cluster Galaxy
BH	Black Hole
BN	Batch Normalization
CNN	Convolutional Neural Network
CoNFIG	The Combined NVSS and FIRST Galaxies catalogue
DDRG	Double-Double Radio Galaxy
DNN	Deep Neural Network
DRAGN	Double Radio sources associated with Active Galactic Nuclei
EMU	Evolutionary Map of the Universe
FIRST	The Faint Images of the Radio Sky at Twenty-Centimeters
FN	False Negative
FNN	Forward Neural Network
FP	False Positive
fpr	false positive rate
FR I	Fanaroff & Riley class I
FR II	Fanaroff & Riley class II
FR II-Low	Low-luminosity FR II object
FR 0	Fanaroff & Riley class 0 or miniature radio galaxy
GLEAM	The GaLactic and Extragalactic All-sky MWA Survey
GPU	Graphic Processing Unit

GRG	**G**iant **R**adio Galaxy
HT	**H**ead **T**ail radio source
HERG	**H**igh **E**xcitation **R**adio **G**alaxy
HYMORS	**HY**brid **MO**rphology **R**adio **S**ource
IC	**I**ndependent **C**omponent
ILSVRC14	**I**mageNet **L**arge-**S**cale **V**isual **R**ecognition **C**hallenge 20**14**
LAS	**L**argest **A**ngular **S**ize
LERG	**L**ow **E**xcitation **R**adio **G**alaxy
LGZ	**L**OFAR **G**alaxy **Z**oo
LINER	**L**ow **I**onisation **N**uclear **E**mission-line **R**egion
LOFAR	**LO**w **F**requency **AR**ray
LoTSS	The **Lo**FAR **T**wo-metre **S**ky **S**urvey
LoTSS DR1	The **Lo**FAR **T**wo-metre **S**ky **S**urvey **D**ata **R**elease **1**
MIGHTEE	**M**eerKAT **I**nternational **GH**z **T**iered **E**xtragalactic **E**xploration Survey
MLE	**M**aximum **L**ikelihood **E**stimation Criterion
MNIST	**M**odified **N**ational **I**nstitute of **S**tandards and **T**echnology database
MOST	**M**olonglo **O**bservatory **S**ynthesis **T**elescope
MSE	**M**ean **S**quare **E**rror
MLP	**M**ulti-**L**ayer **P**erceptron
NAT	**N**arrow **A**ngle **T**ail radio source
NED	**N**ASA **E**xtragalactic **D**atabase
NIN	**N**etwork **I**n **N**etwork
NVSS	The **N**RAO **V**LA **S**ky **S**urvey
PCA	**P**rinciple **C**omponent **A**nalysis
QSO	**Q**uasi **S**tellar **O**bject
RBF	**R**adial **B**asis **F**unction
ReLU	**Re**ctified **L**inear **U**nit
RG	**R**adio **G**alaxy
RGZ	**R**adio **G**alaxy **Z**oo
RGZ DR1	**R**adio **G**alaxy **Z**oo **D**ata **R**elease **1**
RMSProp	**R**oot **M**ean **S**quare **P**ropagation method
ROC	**R**eceiver **O**perating **C**haracteristic curve
SDSS	**S**loan **D**igital **S**ky **S**urvey
SDSS DR 7	**S**loan **D**igital **S**ky **S**urvey **D**ata **R**elease **7**
SDSS DR 12	**S**loan **D**igital **S**ky **S**urvey **D**ata **R**elease **12**
SDSS DR 15	**S**loan **D**igital **S**ky **S**urvey **D**ata **R**elease **15**
SFG	**S**tar-**F**orming **G**alaxy
SMBH	**S**uper **M**assive **B**lack **H**ole
SKA	**S**quare **K**ilometre **A**rray
SGD	**S**tochastic **G**radient **D**escent

SUMSS	The **S**ydney **U**niversity **M**olonglo **S**ky **S**urvey
TGSS	**T**he **G**MRT 150 MHz All-**S**ky Radio **S**urvey
TN	**T**rue **N**egative
TP	**T**rue **P**ositive
VLA	**V**ery **L**arge **A**rray
VLASS	The Karl G. Janksy **V**ery **L**arge **A**rray **S**ky **S**urvey
VAC	**V**alue **A**dded **C**atalogue
WAT	**W**ide **A**ngle **T**ail radio source
WHIM	**W**arm-**H**ot **I**ntergalactic **Medium**
WISE	The **W**ide-field **I**nfrared **S**urvey **E**xplorer
WODAN	**W**esterbork **O**bservations of **D**eep **A**PERTIF **N**orthern sky survey
XAI	e**Xplainable A**rtificial **I**ntelligence
XRG	**X**-shaped **R**adio **G**alaxy

THE UNIVERSITY OF MANCHESTER

Abstract

Faculty of Science and Engineering
Department of Physics and Astronomy in the School of Natural Sciences

Doctor of Philosophy

Giant Radio Galaxies as Probes of the Low Density Intergalactic Medium

by HONGMING TANG

Giant Radio Galaxies (GRGs) as a population are believed to probe the low-density inter-galactic medium. Since their discovery, visual inspection has been the most successful method for GRG candidate selection and radio morphology classification - when the sample size has been manageable. However, visual inspection will no longer be efficient when classifying the millions of objects expected from new generations of radio sky survey. In this case automated classification algorithms then become necessary.

In this thesis I present a transfer learning approach to galaxy morphology classification between different radio surveys which results in models that achieve human-comparable classification accuracy. In addition, I find that inheriting model weights pre-trained on higher resolution survey images (FIRST) can boost model performance when re-training on lower resolution survey images (NVSS). However, the classifier performance deteriorates if this data training sequence is reversed. Consequently, I caution that applying transfer learning when working on new survey data of higher resolution should be carefully undertaken.

I further develop this work to explore CNN-based GRG classification. To start with, I selected source samples from Data Release 1 of the Radio Galaxy Zoo citizen science project. During the sample selection process, I discovered five new GRGs, one of which is also the brightest cluster galaxy (BCG) in a galaxy cluster. I further identified 13 known GRGs as BCG candidates. I examined local galaxy number densities for all known BCG GRGs and found they can reside in the centres of rich galaxy clusters. The existence of this sub-population challenges the GRG formation hypothesis that these galaxies grow to such huge sizes only in low-density environments. With this data set I develop a multi-branch CNN to identify GRGs. This model can learn jointly from both NVSS and FIRST survey images as well as incorporating numerical redshift information. The inclusion of multi-domain survey data improves model performance and corrects 39% of the misclassifications seen from equivalent single domain networks. The inclusion of redshift information moderately improves GRG classification.

Declaration of Authorship

I, HONGMING TANG, declare that this thesis titled, "Giant Radio Galaxies as Probes of the Low Density Intergalactic Medium" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Copyright Statement

Acknowledgements

It's been almost seven years since I came to Manchester and 3.5 years after starting my research work as a Ph.D. student. It was in JBCA I knew my friends in life, my great colleagues, and Lin. Outside JBCA, I am also very grateful to many people worldwide, within or outside the astronomical community, and I would like to present my acknowledgments below.

First of all, I appreciate my main supervisor, Anna Scaife, whom I met as the 'boss' of my summer placement supervisor Minnie. During the years, you are always supportive, helpful, kind, respect my working habit (as my colleagues knew TAKK cafe in Uni. Green is my second or 'main' office). I appreciated that you had guided me during the journey and encourage me to develop my interest (e.g., coffee tasting skills and public engagement). Many thanks again for all of these:)

Besides, I would like to thank the Radio Galaxy Zoo 1 team, including Julie, Ivy, Ray, Anna, Heinz, Stas, Larry, Nick, Chen, Dongwei, and many more. It's such an honor to work with you in the last 3.5 years. I want to thank Julie and Ivy especially. Without your kindness and help, I would not be able to join and work with the team members. Without your support, I can hardly spread RGZ in China, not even advertising citizen science to local citizens. I want to thank you sincerely.

Moreover, I would like to thank my RGZ_CN team members, my colleagues in the Chasing Star public engagement studio, including Chen, Yukun, Hongjie, Tao, Enyu, Jiatong, Yang, and Chuting. I would also like to thank Thijs, the head of IAU East-Asia ROAD. It was my headstrong to found and run this non-profitable citizen science project in the last four years, and all of you have been supporting me with minor complaints. Thanks for valuing the project and the effort of local citizen scientists who got involved.

During the years in JBCA, I have also received a lot of help from my colleagues. I want to say a big thank you to my colleagues in our group: Paddy (my co-supervisor), Katie, Fernando, Alex, Emma, Joe, Fiona, Therese, Rohini, Simon, Micah, David, Miguel, Minnie, Gaargi, and Hayden. I am really happy to work with you, and I will always miss the group. I want to thank Katie, Fernando, Emma, and Joe especially. It's my pleasure to have met you, made friends, and support each other. We have all been through tough times in the last 4, 5, or 6 years, laugh or tears. No matter what happens, I hope we could all stay strong and move on. I wish you all the best and hope we could meet soon sometime in the future.

I also would like to thank my other JBCA and Jodcast colleagues such as Andreas (and Andreas), Shabab, Duncan, Tana, Ant, Andrew, Josh, James, Jack, Laura (thank you for creating this fantastic thesis template!), Shruti, Edorado, Roke, Tiaan, Jianxiong, Bin, Xiaoxi, Xiaojin, Fabian, Mike, Susmita, Tianyue, Issac, Andy, Philippa, Bob, Eunseong, Elizabeth, Naomi, Daniel, Zhuo, Ben, Keith, and Rene. It's a great honor to work with you, and I appreciate the time we work together, chat, and many more.

Before ending the acknowledgment, I would thank Katie, Hannah, Joshua, Matt, Tom, Bruce in the Manchester coffee community. I appreciate you for spending time teaching me how to cup coffee, roast coffee, and even spread the Yunnan speciality coffee to local people. I wish you all running your business well and spread the concept of speciality coffee in greater success. I also want to thank my personal friends in Manchester (i.e., Ke, Xiaoyan, Zihao, Xiaoshuai, Xuzhi, Haoning, Ze, Zhongyu, and Jiayi) and in China (i.e., Sihui, Ge, Ying, Regina, Mian, Xiaojie, Ze, Jiazhen, and Jiahao). I miss all of you, and hopefully, we could meet soon sometime.

Finally, I would like to thank Lin and my parents. Without your support, it would be impossible for me to stay in Manchester and finish my research work at a relatively low-stress level. In the last few years, we met mental or physical health conditions, while we have now gone through all of these, live happily and healthy. Many thanks again, and this thesis is for you.

To Lin and my parents

Chapter 1

Radio Galaxies

In this chapter, I will run through some well-known historic studies of radio galaxies in Section 1.1, and then introduce several current research topics in this field that provide the focus of this thesis.

1.1 Early Radio Galaxy Studies

1.1.1 Cygnus A: The first discovered Radio Galaxy

The study of radio galaxies can be traced back to the 1940s (Bolton & Stanley, 1948; Bolton, 1948). It was at that time that Cygnus A was discovered and identified as the first radio galaxy in history. Early studies of Cygnus A had interpreted the object using pre/post-sequence models (Bolton, 1948), and later accurately positioned and identified it as a radio source (Jennison & Das Gupta, 1953; Baade & Minkowski, 1954). In early 1953, after several pioneering observations made in Cambridge, Manchester (Jodrell Bank) and Sydney (Hanbury Brown et al., 1952; Smith, 1952; Mills, 1952), Jennison and Das Gupta observed the source using the Jodrell Bank Experimental Station, see Figure 1.1, and found that Cygnus A is a double radio source with an angular separation of $1'28''$ (Jennison & Das Gupta, 1953). A contemporaneous study done by Baade and Minkowski, found Cygnus A to be located outside the Milky Way, and to have an optical counterpart. They believed the source to be a pair of colliding galaxies (Baade & Minkowski, 1954).

1.1.2 Early Radio Surveys

In general, radio surveys enable radio astronomers to localize object positions, observe their spectrum, visualize their image at certain frequencies, and finally, produce object catalogues. Radio astronomers then are able to make statistical analyses using these samples, or to select samples from these catalogues to perform further detailed observation (Lukic & Brüggen, 2019).

The earliest celestial radio survey can be traced back to 1940s. Soon after Karl Janksy claimed the first detection of radio waves from outside the Solar System in the 1930s

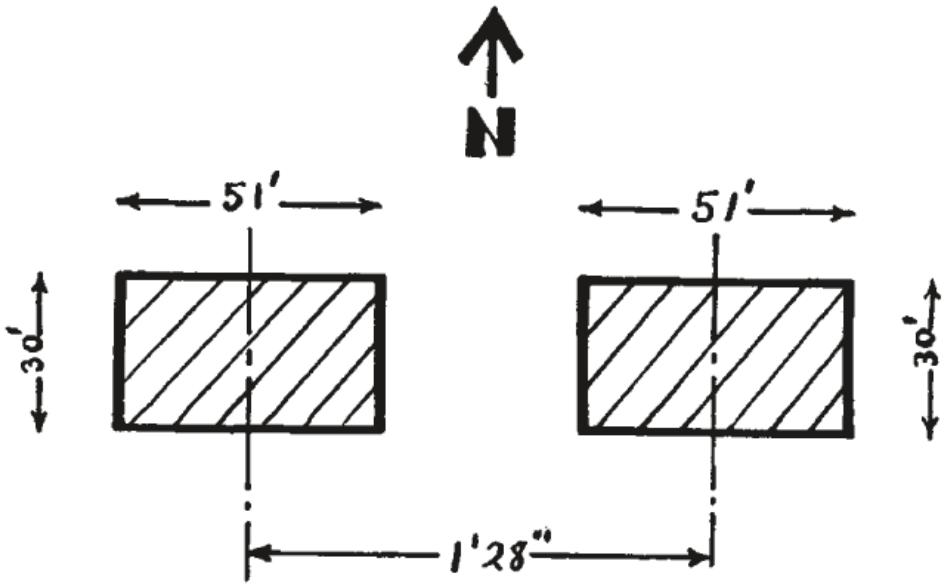


FIGURE 1.1: The approximate intensity distribution of Cygnus A as shown in Figure 2 of Jennison & Das Gupta (1953)

(Jansky, 1933), people started to work on radio sky mapping, or what we call radio sky surveys nowadays. An early attempt was made by Reber in the 1940s (Reber, 1944), using the later named Grote Reber Telescope, see Figure 1.2, a 31.4 foot diameter parabolic radio telescope operating at 160 MHz. The survey observed radio emission from the constellation of Sagittarius, as well as Cygnus, Cassiopeia, Canis Major and Puppis. This early work motivated the later detection of the central radio source in the Milky Way - Sagittarius A* (Piddington & Minnett, 1951; Brown, 1982; Schödel et al., 2002; Gillessen et al., 2012).

Although Reber made significant discoveries with the Grote Reber Telescope, the telescope itself had a limited capacity for identifying radio objects. With a dish diameter of 9.57 m and an observation wavelength, $\lambda = 1.87$ m, we can derive its angular resolution using

$$\theta \simeq \frac{\lambda}{D} \times \frac{180}{\pi}, \tag{1.1}$$

to find that it would have had an angular resolution no better than 10°. Indeed, even the most advanced Five-hundred-meter Aperture Spherical Telescope (FAST; Jiang et al., 2019), with an effective diameter of 300 metres, if operated at the same observational frequency, could only achieve an angular resolution of around 26′. If one wants to observe the radio sky with higher angular resolution, single dish radio telescopes become insufficient to tackle the barrier.

Thankfully, the development of radio interferometry has made this possible. Rather than having D be the dish diameter as for a single dish observation, D for radio interferometry becomes the distance between telescopes, i.e. the baseline length. For instance, the earliest well-known survey that adopted the radio interferometry technology, the First Cambridge catalogue of Radio Sources (1C; Ryle et al., 1950), was operated using

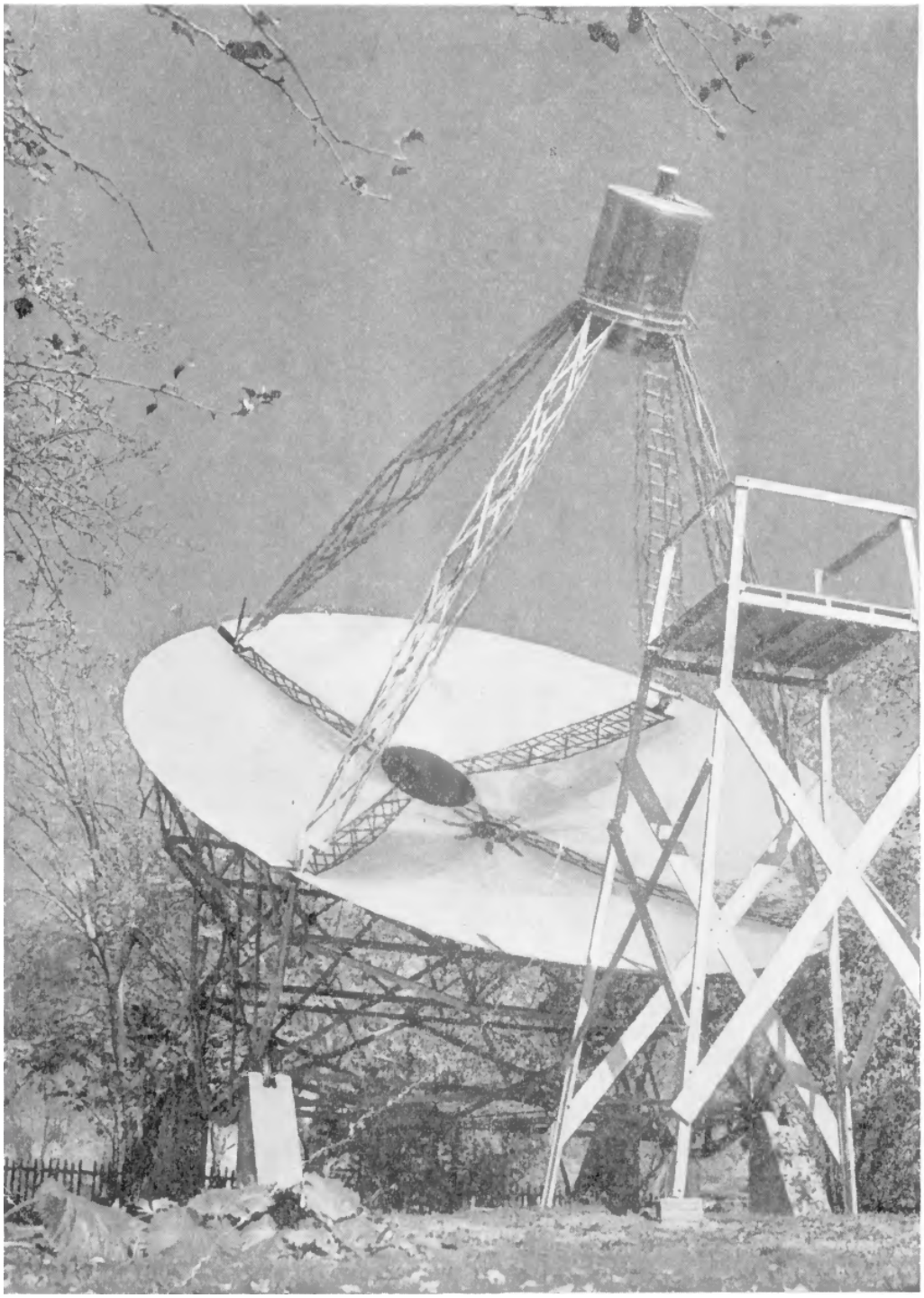


FIGURE 1.2: Photograph of the Grote Reber Telescope, Figure.1 of Reber (1944). The Grote Reber Telescope is a sheet-metal mirror with a diameter of 31.4 feet and a focal length of 20 feet.

an interferometer of two radio antennas called the Long Michelson Interferometer (Ryle et al., 1950). This interferometer had a baseline length equal to 110 times the observation wavelength (3.7 m), and thus an angular resolution of around $31'$. In practice, the interferometer was able to locate most sources to within 1°, while several intense sources could be located with an accuracy of $\sim 5'$. The 1C survey identified 50 discrete sources of radio waves in the Northern sky, and was believed to be extragalactic (Ryle et al., 1950).

Soon after Ryle realised that these sources might be outside of the Milky Way, he and Tony Hewish designed and constructed the Cambridge Interferometer (Ryle, 1952), built to the West of Cambridge. The Cambridge Interferometer contained 4 fixed antennas, and was used for producing the Second Cambridge Survey of Radio Sources (2C; Shakeshaft et al., 1955) at 81.5 MHz and the Third Cambridge catalogue of Radio Sources (3C; Edge et al., 1959) at 159 MHz. Most 3C objects are isotropic and have their luminous radio emissions observed away from the plane of the Galaxy (Edge et al., 1959), and thus are extragalactic. These catalogues contain a few hundred radio sources in the Northern Sky.

Several years after the publication of the 3C catalogue, a revision of the catalogue was made at 178 MHz (3CR; Bennett, 1962b,a). The 3CR catalogue covered all sources with flux densities higher than 9×10^{-26} Watts $\mathrm{m}^{-2}\,(\mathrm{c/s})^{-1}$ at 178 MHz and at $\delta > -5^\circ$ in the 3C catalogue. It was primarily built to further the understanding of the nature of the extragalactic radio sources by measuring the brightness distributions of a large, statistically complete sample (Mackay, 1971). The team selected a large number of sources from the primary version of the 3CR catalogue (Bennett, 1962b,a), measured these objects using the Cambridge One-Mile telescope at 408 and 1407 MHz (MacDonald et al., 1968; Mackay, 1969; Elsmore & Mackay, 1969), and presented their final results in 1971 (Mackay, 1971). Specifically, the 3CR catalogue contains 199 objects mapped at 1407 MHz with a resolution of $23'' \times 23''\mathrm{cosec}\,\delta$, which allowed people to recognize morphological structure in the radio sources. Motivated by the availability of these higher resolution maps, Fanaroff and Riley were able to build a radio galaxy morphology classification system. The system was later known as the FR class (Fanaroff & Riley, 1974).

1.2 Classification Schemes

1.2.1 Fanaroff and Riley Classification

The launch of the FR classification system

Around the time that the final 3CR catalogue was announced, 53 out of the 199 catalogued sources received follow-up observations at 5 GHz with a resolution of $6'' \times 6''\mathrm{cosec}\,\delta$ (Mitton, 1970c,a,b; Graham, 1970; Harris, 1972, 1973; Branson et al., 1972; Riley, 1972, 1973; Riley & Branson, 1973; Northover, 1973, 1974). These high resolution maps allowed people to investigate the source morphologies in order to determine the physical processes behind them.

Fanaroff and Riley selected a subsample of the 3CR catalogue, including only the samples well resolved into 2 or more radio components in any series of the aforementioned observations (Fanaroff & Riley, 1974). By investigating the radio morphology and source luminosities at 178 MHz in $\mathrm{W\,Hz^{-1}\,sr^{-1}}$ of this subsample, Fanaroff and Riley found that the relative positions of the high and low brightness regions in the radio lobes of the galaxies were strongly correlated with their radio luminosities (Mackay, 1971). In practice, they classified objects such that:

"The sources were classified using the ratio of the distance between the regions of highest brightness on opposite sides of the central galaxy or quasar, to the total extent of the source measured from the lowest contour; any compact component situated on the central galaxy was not taken into account."

The ratio parameter, later called the FR ratio, is the 'magic' parameter when doing FR classification: a source with such a ratio of less than 0.5 will be classified as Class I (hereafter FR I), and those sources with FR ratio over 0.5 would be Class II (hereafter FR II). When the angular resolution of an image is good enough, the classification can also be seen as those radio sources with hot spots (the highest brightness regions) closer to their central galaxies as FR I, and those with hot spots further away as FR II (Mackay, 1971).

The key factor that supported the FR classification system and contributed to its later popularity was the distinct difference in source radio luminosities between objects of the two classes: Fanaroff and Riley noticed that, assuming a Hubble constant of $\mathrm{H_0 = 50\ km\ s^{-1}\ Mpc^{-1}}$, almost all their samples with luminosity $L_{178\,\mathrm{MHz}} < 2 \times 10^{25}\,\mathrm{W\,Hz^{-1}\,sr^{-1}}$ were FR I, while those with higher radio luminosities at 178 MHz were FR II. Such difference in luminosity indicates a direct link of radio structure with the way energy transported from the central engine and converted to radio emission in the outer parts (Kembhavi & Narlikar, 1999). Remarkably, this radio luminosity limit is also very close to the limit that divides sources showing strong cosmological evolution (Longair et al., 1973) from those that don't (Fanaroff & Riley, 1974).

Morphological features shared by each class

After Fanaroff and Riley published their classification system, people started to examine its validity on well known radio galaxies. For instance, it was found that Cygnus A is an edge-brightened, classic FR II object, see e.g. Figure 1.3, with the following morphological components clearly visible (Carilli & Barthel, 1996):

- Radio core: the 'central engine' responsible for the double radio source.
- Jets: elongated radio emitting structure connecting the radio core and the source extremities (van Breugel & Miley, 1977).
- Hot Spots: high brightness structure at the extremities of the radio source.
- Radio lobes: the structure in between, usually extended, with low surface brightness and filamentary / diffuse emission.

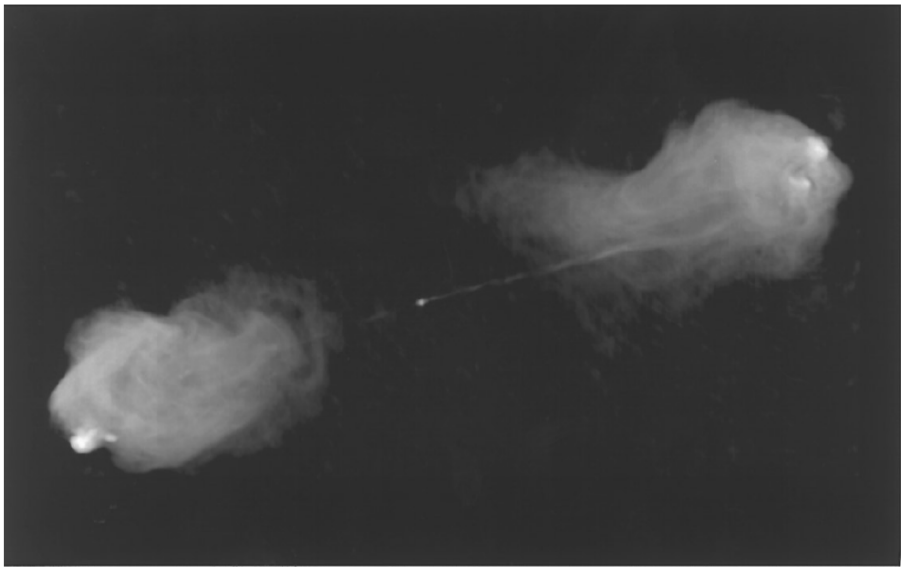


Fig. 2. A greyscale representation of the image of Cygnus A at 5 GHz with 0.4″ resolution made with the VLA (courtesy R. Perley)

FIGURE 1.3: Greyscale image of Cygnus A at 5 GHz, originally shown in Figure 2 of Carilli & Barthel (1996)

These components widely exist among radio galaxies, radio loud quasars and some Seyfert galaxies, which together are also known as Double Radio lobes associated with Active Galactic Nuclei (DRAGNs; Leahy, 1993). In this work, considering the FR classification system, Leahy investigated the source morphology of early discovered DRAGNs (e.g., 1.4), and found that there are some common features in jets shared by either FR I objects or FR II ones.

In the context of jets, according to Bridle & Perley (1984), there are two types of jets connected to a DRAGN:

- Type I: jets of this class generally have wide opening angles (i.e. $5° - 30°$). There exists a gap of around 1 kpc between the central core and a brightening of jet emission. The brightness of the jet then declines smoothly. Also, the twin jets have asymmetric brightness, where the difference is typically a factor of 2. These jets are cross-section brightened.

- Type II: jets of this class share an extremely narrow opening angle. They generally have less regular morphology. The brightness difference between the twin jets is much larger than that of Type I jets. In a few cases, it was found that these jets are 'limb-brightened'.

It was found that DRAGNs with archetypal Type I jets are always FR I objects, while the reverse does not hold true, and Type II jets are found in FR II objects. Moreover, in more than one case, a Type II jet is found in the gap before a Type I jet starts (Leahy, 1993).

On the other hand, the radio lobes of a DRAGN can also be divided into two categories:

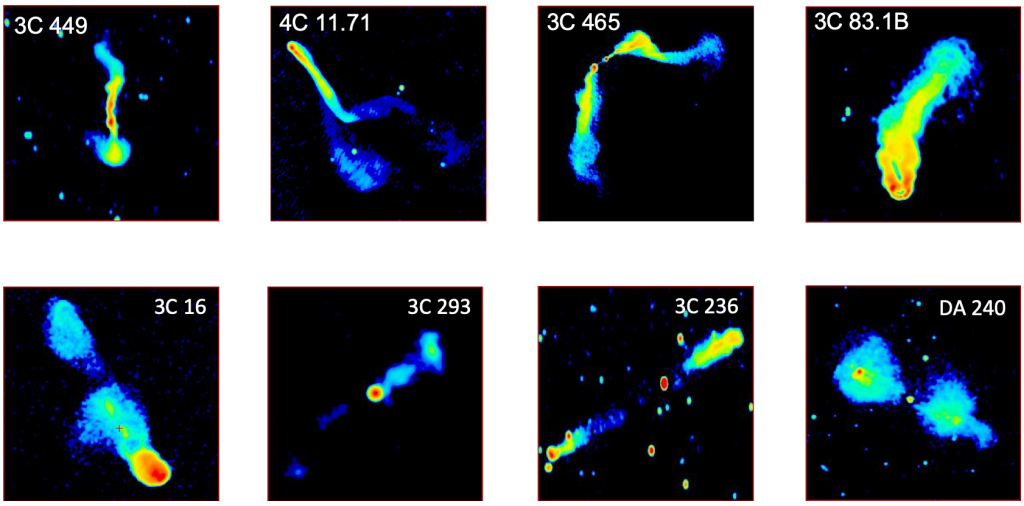


FIGURE 1.4: Example images of FR class identification. The 8 DRAGNs are extracted from the 3CRR catalogue (Laing et al., 1983). These images are displayed using a logarithmic scale, and were retrieved from `http://www.jb.man.ac.uk/atlas/index.html`. Further information on the observation of each source can be found via the webpage. `Upper`: 4 DRAGNs identified as FR Is. Notably: 4C 11.71, 3C 465 and 3C 83.18 have 'Head Tail', 'Wide Angle Tail' and 'Narrow Angle Tail' morphologies, respectively; `Lower`: 4 DRAGNs identified as FR IIs. Notably, 3C 293 is a Double-Double radio galaxy, while 3C 236 and DA 240 are the first two discovered GRGs (Willis et al., 1974).

- Plumes or Tails: plumes or tails are 'jet-like' lobes. The morphology of plumes/tails is largely irregular. The width of these components could be constant, gradually becoming wider, or occasionally become narrow very suddenly. When it comes to brightness differences, the two plumes/tails often show symmetrical brightness. When one refers to a radio component as a plume, it is likely that the component is quasi-linear, and tails are generally connected with jets of Type I or II. In many cases, however, plumes and tails are simply seen as lobes.

- Bridge: a bridge is another type of radio lobe, which fills the gap between the radio core and the hot spots. A bridge was initially defined as diffuse radiation in FR II objects surrounding the presumed path of the jet. It was found that bridges are often found to surround those Type I jets not producing tails (Leahy, 1993).

Furthermore, in terms of general difference, it was found that FR II objects are typically more elongated than FR I sources (Andreasyan et al., 2013).

Are FR Is really different from FR IIs?

As well as the morphological differences, astronomers have also tried to find common physical properties shared by each FR class. For instance, the Owen-Ledlow diagram (Figure 1.5; Owen, 1993; Owen & Ledlow, 1994) is perhaps one of the best known relationships. This diagram visualizes the distribution of a selection of radio sources by plotting source host R-band absolute isophotal magnitude versus source absolute radio power at 1.4 GHz (Owen & Ledlow, 1994). Owen and Ledlow claimed that the morphological FR dichotomy is a function of source radio power and host galaxy optical absolute

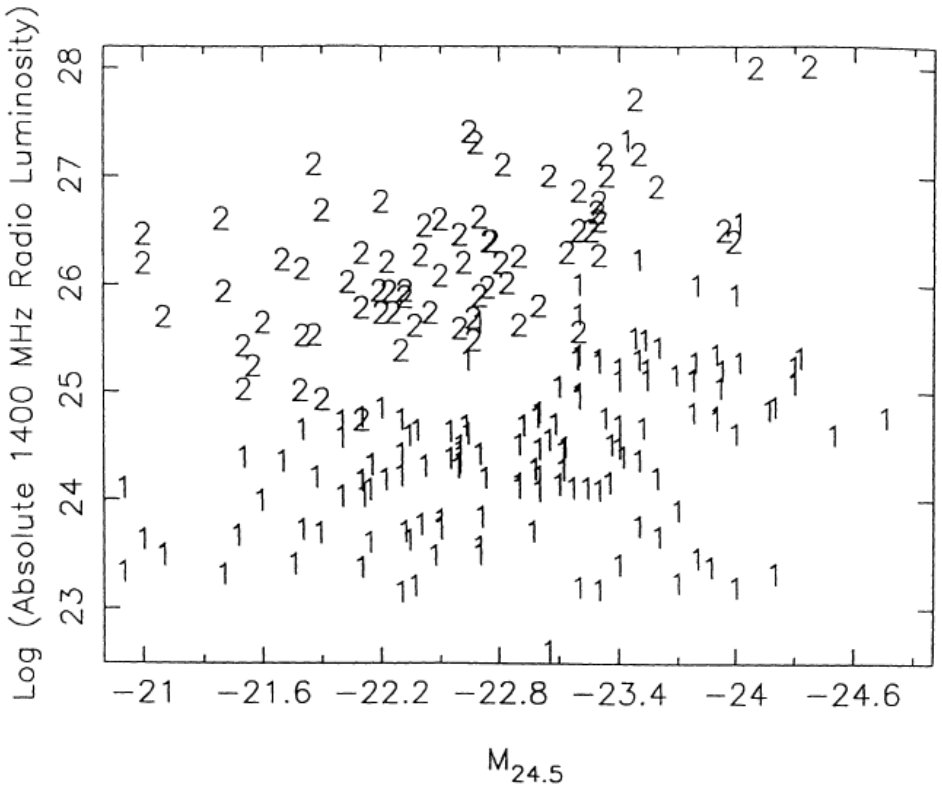


FIGURE 1.5: Figure 1 of Owen & Ledlow (1994), where '1' and '2' on the diagram refer to FR I and FR II class objects, respectively. $M_{24.5}$ refers to the R-band absolute isophotal magnitude of the object host galaxies measured to 24.5 magnitudes arcsec^{-2}.

magnitude. Significantly, the selected objects in the study include 47 objects with host redshift $z < 0.2$ primarily from the 3CRR catalogue (Laing et al., 1983) and another 200 radio sources with cluster redshifts $z < 0.09$ and source radio flux density at 20 cm higher than 10 mJy.

Another scheme related to the FR dichotomy considers the host AGN properties. Conventionally, AGN can be separated into two classes with clear distinctions: High Excitation Radio sources (HERG) and Low Excitation Radio sources (LERG) (Laing, 1994; Heckman & Best, 2014). HERGs are those AGN with strong QSO or Seyfert emission-lines, while LERGs are those with weak Low Ionisation Nuclear Emission-line Region (LINER; Hine & Longair, 1979; Heckman, 1980; Laing, 1994) like emission . In general, LERGs are more massive than HERGs and also host more massive black holes(Heckman & Best, 2014). The radio luminosities of LERG-hosting DRAGNs are generally moderate, while those hosted by HERGs tend to have higher radio luminosity. In 2010, Lin and his colleagues compared host galaxy properties and their corresponding radio source morphologies, and found that while HERGs are predominantly FR IIs, LERGs could host either FR I or FR II type objects (Lin et al., 2010).

On the other hand, in 2012, Saripalli used the Owen-Ledlow diagram as a backdrop and investigated the FR dichotomy (Saripalli, 2012). He found that:

- FR I objects are frequently found to have their radio axis aligned with that of their source host's major axis. Also, in most cases these objects are found to be hosted by more massive elliptical galaxies.
- FR II objects are preferentially hosted by smaller ellipticals or larger ellipticals experiencing mergers.
- LERG FR II objects are found to have circum-nuclear dust and show little active star formation (Baldi & Capetti, 2008), which is similar to FR Is.

Saripalli suggested that FR I objects share more massive hosts probably because they have evolved without much disturbance for a sufficiently long time, and they are then able to re-align with their major axes. The relationship between FR I morphology and massive elliptical host perhaps is benefited from both the high mass of massive host galaxies and their low disk accretion rates. This is because massive elliptical galaxies provide sufficient stellar mass loss to maintain a DRAGN (Di Matteo et al., 2003; Ho, 2009). On the other hand, FR II objects hosted by smaller ellipticals would require mergers as they have insufficient internally generated fuel, and those FR IIs hosted by larger ellipticals require higher accretion rates due to the source internal resistance. It was also believed that at least some LERG FR IIs are transitional FR IIs, dying or restarting. However, it was also pointed out that their properties would require further investigation (Saripalli, 2012).

Interestingly, a few years later, a team of LOFAR project researchers led by Mingo recently questioned the Owen-Ledlow diagram. The evidence they claimed was the discovery of a number of low-luminosity FR IIs in the LOFAR Two-metre Sky Survey (LoTSS;

Shimwell et al., 2017) at 151 MHz. From the LOFAR observations, they found a substantial overlap between FR Is and FR IIs. Unlike the clear luminosity difference claimed by Fanaroff and Riley, more than 20% of the FR II objects found by Mingo's team have luminosities one order of magnitude lower than the original FR radio luminosity dividing line. They then called these sources *FR II-Low* as a bulk sub-classification and found these objects are a heterogeneous population. They were not found when Fanaroff and Riley produced the original FR classification system using the 3CRR catalogue objects as they are both rare in the local Universe and because 3CRR has a comparatively high flux limit (Mingo et al., 2019).

In terms of the formation mechanism for these FR II-Lows, one of the key factors could be age. One of the representative methods to evaluate source age is to calculate source spectral index under assumptions such as constant magnetic field and similar host galaxy redshift (e.g. Blundell & Rawlings, 2001; Harwood, 2017). Spectral index is a measure of the flux density's dependence on frequency, which can be represented by:

$$S_\nu \propto \nu^\alpha, \tag{1.2}$$

where ν refers to the frequency, S_ν is the source flux density at frequency ν and α represents the source spectral index. Source spectral index can be seen as an indicator of the relativistic electron energy distribution. Since electrons would gradually lose energy by radiative processes or non-radiative processes, the spectral index of a source ideally would get steepened as it evolves. Typically, a radio galaxy would have its emission dominated by the optically thin synchrotron emission process, with α of -0.7. By calculating the source spectral index of these FR II-Low objects, Mingo and her colleagues found that a subset of these sources have spectral index $\alpha < -1.0$. This implies that at least some of the FR II-Low sources are probably older sources. On the other hand, the rest of the FR II-Low samples have α values between -1 and -0.7, which coincides with the same distribution for FR II-High, and thus age cannot be the only contributing factor to explain FR II-Low objects (Mingo et al., 2019).

Besides age, the team also compared the host luminosity distribution between their FR II-Lows and FR Is of similar radio luminosity (Mingo et al., 2019). Assuming that jet disruption models (e.g. Bicknell, 1995; Kaiser & Best, 2007) hold for the FR dichotomy, the disruption of jets might be caused by the interaction between the jet power with the environmental density (Kaiser & Best, 2007; Mingo et al., 2019). In this case, objects with similar jet powers close to the FR break would be more likely to get disrupted if their hosts' surrounding ambient medium is denser. These jets then would be more likely to reach pressure equilibrium, lose their protective lobe and develop into FR I objects (Kaiser & Best, 2007). FR II-Lows with similar jet power should then have their environmental densities differ from those of FR Is. Assuming that host galaxy optical luminosity is a good proxy of galaxy local density, Mingo's team compared the host galaxy luminosity between FR II-Lows and FR Is, they found the host optical luminosity of FR II-Lows is systematically lower than that of FR Is. They then concluded that the other possibility for

FR II-Low object formation could be sources beaming low-power jets when their hosts are of lower mass compared with the FR IIs of higher luminosity or FR Is of similar luminosity, consequently the jets of these FR II-Lows therefore could remain undisturbed. The team did not investigate the correlation between the FR II-Lows and excitation as they lacked source excitation information.

Although the discovery of FR II-Lows has challenged the classical FR definition, the FR dichotomy still shows some significant unexplained differences in source jet/lobe morphology, as well as host properties. It therefore continues to be necessary to perform FR classification on newly identified DRAGNs, in order to study the properties and formation mechanisms of each class.

1.2.2 Beyond FR: DRAGNs with irregular morphology

As long as people continuously study DRAGN morphology, more radio sources with unusual morphologies are discovered. In this section, I will run through the basics of the most representative species of DRAGNs with unusual morphologies. Those radio sources that belong to a minority defined according to their host properties (i.e., Spiral DRAGN (e.g. Hota et al., 2011; Bagchi et al., 2014); Radio Peas (e.g. Chakraborti et al., 2012)) or those that are not correlated to AGN (i.e., Radio Gischt (e.g. Kempner et al., 2003); Radio Halo (e.g. Liang et al., 2000)) are not considered here.

I separate these species into three categories: (a) 'bent tailed' objects; (b) FR system supplements and (c) Irregular sources.

a.1 Head Tail sources

The study of 'Head Tail' (HT) radio sources can be traced back to 1968, when NGC 1265 was discovered as the first HT object (Ryle & Windram, 1968). A few years later, Miley proposed the name 'Head Tail' to describe the subclass that describes these radio sources (Miley et al., 1972). This type of object was originally believed to be unusual, while people now believe they are very common (Mao et al., 2009). Generally, a 'Head Tail' radio source will have its jets bent back to form 'tails', and the bright host galaxy is seen as the 'head'.

In the context of source properties, most HTs are believed to be FR Is and live in galaxy clusters (Komissarov, 1988). A later study done by Mao and her colleagues further showed that HTs live in denser environments compared to other radio sources and among the densest in clusters (Mao et al., 2009). She further proposed that it is the high densities that allow ram pressure to bend HT objects (Mao et al., 2009). Interestingly, recent studies have claimed that some HTs with narrow tails are not different from NATs (see following subsection). They are likely to be visually edge-on, and strongly affected by projection effects (Terni de Gregory et al., 2017).

a.2 Wide Angle Tail

Wide Angle Tail (WAT) radio sources were first observed by Owen and Rundick in 1976 during a survey of radio sources in Abell galaxy clusters (Owen & Rudnick, 1976). The name WAT comes from the morphological feature where their jets are bent in a common direction, with extended plumes, and have visible hotspots closer to the central active galaxy than that of FR IIs (Mao et al., 2010). WATs were originally argued to be an extension of the HT subclass when discovered, as they share large opening angles between their tails (Owen & Rudnick, 1976). These days, the WAT population includes both HT and those objects with opening angles greater than 90° (Rudnick & Owen, 1977; Mingo et al., 2019). WATs are mostly found in galaxy clusters, and thus are usually seen as galaxy cluster tracers at moderate or higher redshifts (Mao et al., 2010). The jet bending of WATs is commonly attributed to strong intracluster winds, thought to be caused by cluster-cluster mergers (e.g. Burns, 1998; Mao et al., 2010).

In terms of the FR dichotomy, WAT objects are generally identified as FR Is, and they are believed to be a sub-population of FR Is (e.g. Mingo et al., 2019). A recent study investigated 47 WATS and found that they lie in the region where FR Is are most densely populated in the modified Owen-Ledlow diagram (Missaglia et al., 2019). The WAT sample was LERG, with remarkably homogeneous host galaxy properties, and IR colors similar to that of FR Is. Moreover, the team found that more powerful WATs tended to be hosted by more massive galaxies. However, by looking at the modified Owen-Ledlow diagram, the team also recognized that their WATs follow the same behaviour as HERG FR IIs and have radio powers typical for FR IIs. Whether it is appropriate to simply see WATs as a subset of FR Is or to identify them as another separate class still requires further investigation.

a.3 Narrow Angle Tail

In 1977, Rudnick and Owen first used the terminology 'Narrow Angle Tail' (NAT; Rudnick & Owen, 1977). They performed interferometric observations towards a sample of radio sources in Abell galaxy clusters, and found that over 2/3 of their sample were distorted or showed misalignment. These sources were then labelled as narrow, intermediate and wide angle tailed sources (Rudnick & Owen, 1977), depending on the opening angle between the hot spots and the host galaxy. NATs were originally defined as those objects with their opening angle smaller than 20°. More recent NAT definitions have merged the NAT with the 'intermediate' angle tail objects and now only requires the opening angle to be smaller than 90° (Mingo et al., 2019).

The formation of NATs is believed to be the result of ram pressure from either the relative motion of an AGN host through its surrounding environment, or 'weather' motion within such environment, or both, producing pressure forces and deflecting AGN jets (Begelman et al., 1979; Jones & Owen, 1979; O'Neill et al., 2019). Recently, a team led by O'Neill performed an simulation-based study of NAT dynamics. They claimed that although they confirm the traditional jet bending models, they also found that the

formation process of NATs might have include a transient stage in its early phase, when the jets are bent mildly making the source look like a WAT (O'Neill et al., 2019).

b.1 HYMORS

HYbrid MOrphology Radio Sources (HYMORS) are a class of DRAGN with one side showing FR I-type lobes, and the other side showing FR II-type lobes (Gopal-Krishna & Wiita, 2000; Gawroński et al., 2006). The discovery of HYMORS was difficult from early surveys, with a less than 1% occurance rate in FIRST images at 1.4 GHz. However, thanks to the LOFAR telescope's capability for identifying faint and old objects, project scientists found a HYMORS incidence of 25% from the catalogue of LoTSS extended sources (Mingo et al., 2019).

In terms of formation mechanism analysis, researchers value HYMORS as their existence could help to disentangle two aspects of the FR dichotomy:

- (i) Nature: the FR dichotomy might be caused by a fundamental difference in their central engines.
- (ii) Nurture: the FR dichotomy could be explained in the context of relativistic outflow power, deceleration and jet interaction with the local environment.

From an early sample of HYMORS, Kapinska and her colleagues suggested that these objects could be formed via either of these mechanisms and could be hosted by any kind of active galaxy (Kapińska et al., 2017). In recent years, although it is still under debate which mechanism could drive HYMORS formation, people have found that they might be heterogeneous and projection effects become a factor difficult to eliminate during such studies (Mingo et al., 2019). Moreover, the most recent study in this field has claimed that HYMORS morphology is most likely to be caused by projection effects, and that they are intrinsically FR IIs. These objects tend to have bent jets, lobes that are non-linear to their corresponding jets, or both (Harwood et al., 2020).

b.3 FR 0

The term 'FR 0', or 'miniature radio galaxy' in earlier studies, refers to compact radio galaxies (corresponding to linear sizes ≤ 10 kpc) found in the local Universe (Baldi & Capetti, 2009; Ghisellini et al., 2011; Sadler et al., 2014; Baldi et al., 2015). It was found in a pilot JVLA project that there exist a large number of compact or slightly resolved radio sources on scales of a few kpc. These objects are found to be hosted by early-type red galaxies with high central blackhole masses: $10^8 - 10^9 M_\odot$, which is similar to that of FR Is (Baldi et al., 2015, 2018). In early FR 0 studies, the only distinct difference people found between FR 0s and FR Is was the unresolved morphology of FR 0s.

In order to better investigate FR 0 properties in bulk, Baldi's team selected a sample of 108 radio loudAGN from a much larger catalogue (Best & Heckman, 2012) called FR0CAT (Baldi et al., 2018), requiring each sample to have a host galaxy redshift less

than 0.05 and with a radio size ≤ 5 kpc (Baldi et al., 2018). By investigating FR0CAT samples with high resolution observations, the team found that these FR 0s were unresolved in FIRST images, and are spectroscopically defined as LERGs (Baldi et al., 2019). Later studies using FR0CAT samples also found that they reside in less dense environments compared to FR Is, suggesting that there exists a connection between environment and jet power. Such a connection is driven by a common link with blackhole spins (Capetti et al., 2020a). Interestingly, while earlier studies saw FR 0 and FR I as two distinct populations, recent LOFAR observations upon a subset of the FR0CAT samples found that at least 40% of their sample revealed evidence of jet structure existence. From this result, the study finally interpreted FR 0s and FR Is as the two ends of a continuous population of jetted sources with FR 0s at the lower end of the radio power and size distribution (Capetti et al., 2020b).

c.1 X-RG

X-shaped Radio Galaxies (XRGs), or 'wing-like' radio galaxies are a rare class of radio source with a peculiar morphology (Leahy & Williams, 1984; Worrall et al., 1995). These objects share wing-like structures, and in some cases show 'double boomerang' morphology (Leahy & Williams, 1984; Cotton et al., 2020). The class definition of XRGs comes from their second set of jets or 'wing-like' structures, which are misaligned with the first ones (Leahy & Williams, 1984; Cotton et al., 2020). Since 3C 315 was recognized as one of the first XRGs in the 1970s (Hogbom & Carlsson, 1974), there have been three dominant models of XRG morphology (Cotton et al., 2020):

(i) The source central SMBH has experienced a sudden or continuous reorientation, especially caused by BH-BH merger (Zhang et al., 2007).

(ii) The X-shaped lobes are the superposition of two set of jets beamed from two SMBHs living in the same host galaxy (e.g. Lal & Rao, 2005)

(iii) The 'wing-like' structures come from the hydrodynamical backflows of the source main lobes (Leahy & Williams, 1984; Worrall et al., 1995; Hardcastle et al., 2019a).

Although the debate on how XRGs are formed has existed for decades, recent observations made by the SKA pathfinders LOFAR and MeerKAT seem to challenge Models (i) and (ii) above (Hardcastle et al., 2019a). Thanks to the high resolution and high sensitivity of LOFAR, recent observations of NGC 326 (Figure 1.6) found that the source image can no longer be interpreted using the archetype of Model (i) (Hardcastle et al., 2019a). They found that NGC 326's wings to extend into gigantic tails, requiring a hydrodynamic explanation since the source radio structure has been bent significantly. Model (i) itself then becomes insufficient to explain the phenomena.

The most recent observation, made by MeerKAT, on the other hand, has found some key evidence to support Model (iii). By observing PKS 2014-55, the team led by Cotton found the source image to show clear and continuous hydrodynamic backflow from a

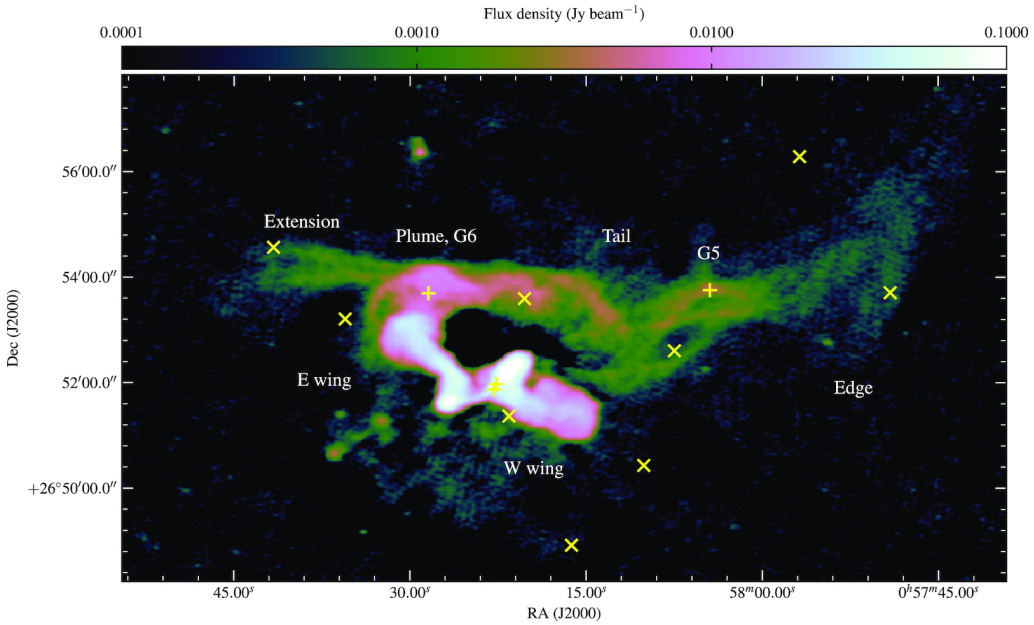


FIGURE 1.6: Figure 1 of Hardcastle et al. (2019a). The figure shows the logarithmically scaled radio emission observed by LOFAR. The color bar is in units of Jy beam^{-1} for the observational resolution of $8.2'' \times 5.1''$. Yellow crosses on the diagram mark the potential cluster members; + signs refer to the spectroscopic study of Werner et al. (1999), and $\times$ signs mark the candidate galaxies described in Hardcastle et al. (2019a). The double-nuclei of NGC 326 are shown as adjacent crosses at the centre of the diagram. Morphological features of the source are labelled on the diagram in white.

pair of collimated jets with fixed orientation redirected by the host galaxy hot ISM (Cotton et al., 2020), which disfavours Model (ii). This observation has also noticed a restarted jet in the same direction as the main lobes, which largely disfavours Model (i). In all, though these observations cannot explain the formation mechanism of XRGs as a population, they have proved that telescope arrays with high resolution, good brightness sensitivity and diverse baselines like LOFAR and MeerKAT have the potential to tackle the nature of XRGs.

c.2 Double-Double Radio Galaxies

Double-Double Radio Galaxies (DDRG; Schoenmakers et al., 2000b) are radio galaxies that have a pair of double radio hotspots with a common centre, and have inner lobes with a clear edge-brightened FR II morphology. The origin of such source inner lobes is most likely to be restarted jets, and the time required to restart should be shorter than that required for the outer lobes to fade away and become too faint to be observed on radio maps (e.g. Schoenmakers et al., 2000b; Jurlin et al., 2020).

In the context of source extent, early detections claimed that DDRGs have linear sizes over 700 kpc (Schoenmakers et al., 2000b), while later studies have found these sizes to vary from less than 1 kpc to over 4 Mpc (Saikia & Jamrozy, 2009; Nandi & Saikia, 2012; Kuźmicz et al., 2017). Other than size, an interesting topic in the field is the formation mechanism of these remnant and restarted objects: Kuźmicz et al. (2017) claimed that the restarted objects had smaller concentration indices compared with classical FR IIs, which

might be the outcome of frequent merger events in the history of their host evolution. However, Mahatma et al. (2019) and Jurlin et al. (2020) have a different point of view on this problem. They investigated the radio and optical properties of radio sources in difference phases of their life cycle, and found these groups to have no difference in optical and radio properties. This implies that DDRGs are not the consequence of such host galaxy changes, and that the two phases (DDRG and classical FR IIs) are actually coming from the same parent population (Jurlin et al., 2020). However, this is an ongoing research field, and further observations will perhaps solve the formation issue.

c.3 Relic Radio Galaxies

Relic radio galaxies are a rare class of DRAGNs with only radio lobes visible. Their core, jet and hotspots have disappeared since they have passed their active phases (Murgia et al., 2011). These objects may either have no AGN activity or activity at such a low level that outflowing jets cannot be sustained any more (Tamhane et al., 2015). The rarity of these relic radio galaxies could be explained by their short period of visibility ($10^6 - 10^7$ yr), which is caused by radiative losses (Murgia et al., 2011). In other words, relic radio galaxies could be seen as a final phase of radio source evolution (Tamhane et al., 2015).

The discovery of these relic sources can be traced back to the 1980s. Until 2015, only a few relic sources were known (e.g. Cordey, 1987; Venturi et al., 1998; Murgia et al., 2011; Hurley-Walker et al., 2015; Tamhane et al., 2015), including J021659-044920 as a relic Giant Radio Galaxy (GRG; Willis et al., 1974) . These sources are found to have steep spectrum (i.e. spectral index $\alpha \leq$ -1), which makes their emission difficult to detect at high frequencies. The emergence of LOFAR and the SKA at low frequencies perhaps could facilitate future relic radio galaxy discoveries (Kempner et al., 2003; Murgia et al., 2011).

In the context of source spectral index studies, recent observations of J021659-044920 between 0.325 MHz and 1.4 GHz have found that the radio lobe spectral indices experience gradual steepening from the outer part of the lobes to the inner regions (towards the source host). This phenomena is consistent with the backflow model, where the pressure of jets will re-accelerate post-shock and cause backflows toward the source core. In other words, the inner radio lobes are backflowed and thus have steeper spectral indices. Further studies towards these rare objects will require imaging with high sensitivity, high resolution over multiple frequencies.

1.3 Giant Radio Galaxies

Aside from radio galaxy morphology, the dominant focus of this thesis is a group of comparatively rare DRAGNs, known as Giant Radio Galaxies (GRGs). They are the largest radio galaxies in the Universe. Originally defined to be those radio galaxies with projected linear sizes greater than 1 Mpc in a cosmology with $H_0 = 50$ km s^{-1} Mpc^{-1} (Willis

et al., 1974), the GRG size limit is now equivalent to 700 kpc in a ΛCDM cosmology with the Planck 2016 parameters (Planck Collaboration et al., 2016; Dabhade et al., 2020a).

The primary motivation for finding such giant radio sources is to investigate the possible modes of energy replenishment that allow for the existence of such a population (Longair et al., 1973), as energy losses over the physical scales associated with these gigantic radio components are unavoidable. Some believe that the gigantic size of GRGs might be caused by high kinetic jet power (Wiita et al., 1989) and it has been shown that the size of a radio source is positively correlated with source radio luminosity and jet power (Shabala & Godfrey, 2013). Alternatively, it has also been proposed that GRG sizes might be caused by the comparative longevity of their jets (Subrahmanyan et al., 1996), or due to the radio source growing in a low density environment (Malarecki et al., 2015).

Thanks to the comparatively large angular extent of the GRG population, astronomers can observe their fine structures with detailed imaging, and can therefore assist in studies of the physical processes occurring within the galaxies themselves (Willis et al., 1974). This idea has motivated an interest in the discovery of GRGs with unusual morphology. This includes the discovery of several giant Double-Double Radio Galaxies (DDRG; Schoenmakers et al., 2000b; Saikia et al., 2006; Bagchi et al., 2014) implying that jet-interruption might have taken place within radio sources with such unique radio morphology. Moreover, Solovyov & Verkhodanov (2011, 2014) reviewed a list of GRG candidates, and found 8 XRGs with signatures of galaxy interaction. And finally, people have discovered two HYMORS GRGs, allowing astronomers to study the formation mechanism of this species in more details. To date, there are over 800 GRGs identified in the literature (Kuźmicz et al., 2018a; Dabhade et al., 2020b,c; Tang et al., 2020; Delhaize et al., 2021), and the work of hunting for new GRGs is still ongoing.

The role of local environment in GRG formation was primarily pointed out by Ishwara-Chandra & Saikia (1999) who compared 53 GRGs in the literature with 3CR radio sources (Laing et al., 1983) of smaller sizes. Ishwara-Chandra & Saikia (1999) found that GRGs share a marginally higher separation ratio of hotspot distances from the nucleus, which Ishwara-Chandra & Saikia (1999) suggest might be caused by the interaction of energy carrying beams and cluster-sized density gradients far from the source host galaxy. This has in turn led to GRGs being used as probes of the low ambient density warm-hot intergalactic medium (WHIM; Safouris et al., 2009; Peng et al., 2015).

Another environmental consideration is the local galaxy density around GRGs. GRGs have typically been found in under-dense environments, and it has been proposed that such reduced galaxy densities facilitate these radio galaxies to grow larger (Malarecki et al., 2015). However, a number of other studies have found that there is no correlation between radio source linear size and local galaxy density (Komberg & Pashchenko, 2009; Kuźmicz et al., 2018b; Ortega-Minakata et al., 2018). Moreover, the recent discovery of more than 20 GRGs that not only reside in galaxy cluster environments but are also the brightest galaxy in these clusters (brightest cluster galaxies; BCGs) has also challenged this hypothesis (Dabhade et al., 2017, 2020a; Tang et al., 2020).

From a galaxy evolution perspective, GRGs represent the tail of the radio galaxy size distribution. A comprehensive study of the shape of this distribution requires consistent sampling of both GRGs and smaller radio galaxies. However, traditional methods of cross-matching large scale radio surveys, like the Faint Images of the Radio Sky at Twenty-Centimeters (FIRST; Becker et al., 1995) survey, with optical/infrared surveys such as those obtained using the Wide-field Infrared Survey Explorer (WISE; Wright et al., 2010), e.g. the AllWISE image atlas and catalogue (Cutri & et al., 2013), are complicated by scale-dependent observational selection effects, as well as the uncertainties in cross-matching which arise when dealing with diffuse or complex radio emission.

Citizen science offers an alternative to more traditional methods of building large cross-matched radio galaxy catalogues. Radio Galaxy Zoo (RGZ; Banfield et al., 2015) is an online citizen science project which aims to cross-match extended radio sources from the FIRST survey (Becker et al., 1995) and the Australia Telescope Large Area Survey (ATLAS; Franzen et al., 2015) with their host galaxies in the infrared waveband, using data from the AllWISE survey and the SIRTF Wide-Area Infrared Extragalactic Survey (Lonsdale et al., 2003). RGZs offers its volunteers a 3×3 arcmin^2 cutout from the FIRST survey with radio contours starting at $3\sigma_{rms}$ on top of a WISE 3.4 μm image. Project participants are asked (a) to identify radio components of a source from an image, (b) to select the infrared host galaxy of the corresponding radio source, and (c) to check if there are additional sources without existing identifications present in the same image (Banfield et al., 2015). The project is intended to provide the foundation of a large cross-matched radio galaxy catalogue.

1.3.1 Hunting GRGs: General Guidelines and Popular Approaches

In general, the physical size of a source can be calculated if its host galaxy redshift (z) and its largest angular size (LAS) are available. Regardless of exact methodology, classifying a radio galaxy as a GRG requires astronomers to (1) identify radio components belonging to the same DRAGN; (2) find the corresponding host galaxy of the DRAGN; (3) measure the source's Largest Angular Size (LAS) from radio maps; (4) measure the host galaxy redshift, and (5) derive the source's projected physical linear size based on the source LAS and host galaxy redshift.

Step 1-2: Identifying DRAGNs

Among the five steps of GRG identification, step 1 and 2 perhaps are the most time-consuming part of the work. Traditionally, a limited number of experts would first identify radio source components and then cross-match their optical/infrared hosts (Subrahmanyan et al., 1996; Lara et al., 2001a; Machalski et al., 2001; Schoenmakers et al., 2001; Saripalli et al., 2005; Machalski et al., 2007; Solovyov & Verkhodanov, 2011). The way to identify DRAGNs is known as visual inspection. In order to identify a DRAGN, experts need to manually look at radio maps and search their optical/infrared counterparts

with their priori knowledge. Further information of hunting GRGs via visual inpsection would be described in Section 1.3.1.

Rather than pure visual inspection, such work could be done by semi-automated source matching. Recently, thanks to the availability of large optical and radio surveys, Dabhade et al. (2020a) discovered 225 new GRGs using the Value Added Catalogue (VAC; Williams et al., 2019) of the LOw Frequency ARray (LOFAR; van Haarlem et al., 2013). Most compact sources in the VAC catalogue are selected by cross-matching the LOFAR Two-metre Sky Survey Data Release 1 catalogue (LoTSS DR1; Shimwell et al., 2017, 2019) with a catalogue of matches between the Panoramic Survey Telescope and Rapid Response System (Kaiser et al., 2002, 2010; Chambers et al., 2016) catalogue and the AllWISE catalogue, using a likelihood ratio method (Williams et al., 2019; Dabhade et al., 2020a). However, diffuse and complex sources in the catalogue are cross-matched by visual inspection via a private Zooniverse project - LOFAR Galaxy Zoo project (Williams et al., 2019). Among the 231,716 sources of LoTSS DR1 that have optical/IR identifications, team project scientists found there are only 0.1% of GRGs (Williams et al., 2019; Dabhade et al., 2020a).

Besides the empirical or semi-automated approaches performed by professional astronomers, citizen science have also shown its power. The aforementioned project RGZ (Banfield et al., 2015) is an representative one. A key advantage of RGZ is that it requires its radio and infra-red image data to share a uniform 3×3 arcmin field of view. Thanks to the uniform data image size, project team scientists are able to develop an end-to-end automated radio galaxy detection/classification algorithm ClaRAN (Wu et al., 2019). Though imperfect, deep learinng algorithms like ClaRAN could be seen as the fourth path of DRAGN identification, and have the potential to tackle the barrier of big data challenge when dealing with the new radio sky continuum surveys.

Step 3-5: Expert visual inspection

The majority of historic GRG studies use 'by eye' classification, also known as visual inspection. In these studies, new GRGs were confirmed by following-up sample candidates pointed out by previous studies, or by searching large-scale radio survey catalogues by eye (Willis et al., 1974; Bridle et al., 1976; Waggett et al., 1977; Laing et al., 1983; Kronberg et al., 1986; de Bruyn, 1989; Jones, 1989; Ekers et al., 1989; Lacy et al., 1993; Law-Green et al., 1995; Cotter et al., 1996; McCarthy et al., 1996; Subrahmanyan et al., 1996; Ishwara-Chandra & Saikia, 1999; Schoenmakers et al., 2000a; Lara et al., 2001b; Machalski et al., 2001; Sadler et al., 2002; Letawe et al., 2004; Saripalli et al., 2005; Saikia et al., 2006; Machalski et al., 2008; Huynh et al., 2007; Machalski et al., 2008; Kozieł-Wierzbowska & Stasińska, 2011; Hota et al., 2011; Solovyov & Verkhodanov, 2011; Molina et al., 2014; Solovyov & Verkhodanov, 2014; Bagchi et al., 2014; Amirkhanyan et al., 2015; Tamhane et al., 2015; Amirkhanyan, 2016; Dabhade et al., 2017; Clarke et al., 2017; Kapińska et al., 2017; Prescott et al., 2018; Sebastian et al., 2018; Koziel-Wierzbowska et al., 2020a; Dabhade et al., 2020b,c; Tang et al., 2020; Delhaize et al., 2020).

Before large scale radio surveys such as NVSS or FIRST were available, most studies in this field would examine the validity of a GRG candidate by making a deep observation of a particular source or set of sources in specific radio and optical wavebands and would use the optical spectrum of the galaxy host in order to measure source redshift (Bagchi et al., 2014; Amirkhanyan et al., 2015; Tamhane et al., 2015). When large-scale radio surveys (e.g. NVSS, SUMSS, FIRST), and optical surveys such as the Sloan Digital Sky Survey (SDSS; Albareti et al., 2017) became available, later studies tended to select source candidates from a particular survey and to perform early cross validation using other survey image data if available. With the availability of photometric redshifts for large numbers of objects from surveys such as SDSS, a number of more recent discoveries have also used photometric redshift when estimating object distances (Dabhade et al., 2020b,c; Tang et al., 2020). Such approaches have enabled researchers to measure source host redshift and object LAS of these GRGs with excellent reliability, and investigated their spectral properties (Dabhade et al., 2020b).

Step 3-5: Citizen Science Facilitation

Although RGZ as mentioned was initially launched to create a large scale radio galaxy catalogue with the help of citizen scientists, its online forum *RadioTalk* allowed citizen scientists to collaborate with project team scientists and find radio galaxies of special types. Citizen scientists pointed out the GRG candidates on the *RadioTalk* forum and notified project scientists. Project scientists then cross-validate the candidates with multi-frequency archival data (e.g. NVSS, SDSS, WISE, XMM-Chandra). In fact, six out of the eight radio galaxies confirmed as GRGs by the RGZ team were pointed out on *RadioTalk* as GRG candidates by several project citizen scientists in advance of their confirmation (Banfield et al., 2016; Kapińska et al., 2017; Tang et al., 2020). Chapter 4 will explain the method in more details using my recent research as an example (Tang et al., 2020).

Step 1 and 3 only: Decision Tree Approach

The two aforementioned methods are widely accepted as they allow for cross validation with diverse complementary radio/optical survey images, deep source imaging and/or spectral confirmation. Also, the involvement of prior knowledge of DRAGNs have largely increased the consensus level of such approaches. However, while such approaches work well when dealing with source catalogues such as NVSS, SUMSS and FIRST, with modest sample sizes, such methods become impractical when faced with millions rather than thousands of candidate galaxies. For example, the Evolutionary Map of the Universe (EMU; Norris et al., 2011) survey, one of the the Australia SKA Pathfinder (ASKAP; Johnston et al., 2008) early science projects, is expected to provide a catalog containing approximately 70 million sources, of which $\sim$ 7 million will require visual inspection. Reliable automated GRG search methods therefore become necessary in the era of astronomical big data.

One recent GRG search method Proctor (2016) attempted to challenge the traditional visual inspection methods by using a decision tree based machine learning approach. The study focused on sources with only two radio components and aimed to identify GRGs among them, regardless of component morphology (lobes, jets, core). The training set for this work consisted of 51 195 source pairs from the NVSS catalogue, of which 48 had previously been confirmed to be GRGs by Lara et al. (2001b). When selecting sample features, the study mostly used component major axis, minor axis and peak flux as input features. The best classifier in this study achieved a training accuracy of 97.8±1.5%. Proctor (2016) then used the best classifier to find GRGs from 870 000 candidate pairs in the full NVSS catalogue, extracting those objects with high GRG probabilities. Such a semi-automated procedure produced a list of 1 616 GRG candidates with LAS $\geq 4'$.

Although this pioneering study predicted a large number of GRG candidates, it did not consider source host galaxy redshift as this was unavailable in the NVSS catalogue. In other words, this study has only concentrated on sorting out step 1 and 3 with strict selection rules. Consequently, only 16 of the selected candidates were included in the later updated catalogue of GRGs (Kuźmicz et al., 2018a). This catalogue assumes H_0 = 71 km s^{-1} Mpc^{-1}, Ω_M=0.27, $\Omega_{\rm vac}$=0.73, and lists 349 confirmed GRGs as of the end of 2018. The catalogue uses host galaxy redshifts either from the literature or from SDSS.

Limited by the lack of host galaxy cross-identification and their redshift information, validation of the sample from Proctor (2016) was not comprehensively addressed until Dabhade et al. (2020c) performed a follow up study. Thanks to the availability of source coordinates in the Proctor (2016) candidate catalogue, the team was able to track positions for each candidate, manually visualize their NVSS, FIRST, The GMRT 150 MHz All-sky Radio Survey (TGSS; Intema et al., 2017), and the Karl G. Jansky Very Large Array Sky Survey (VLASS; Lacy et al., 2020) images (if available), and also check their host galaxy redshift from publicly available optical surveys and databases (Dabhade et al., 2020c). Source LAS were measured from NVSS images for uniformity, where only emission above a 3σ level was considered. The team found that there were 165 known and 151 newly discovered GRGs among the candidates. In other words, around 20.8% of the candidates in the list were finally confirmed to be GRGs.

Although the algorithm of Proctor (2016) did not predict a fully reliable set of GRGs from the test sample, it had successfully produced a good candidate pool. The availability of traceable source coordinates allowed people to follow and finish the GRG hunting guidelines. People then are able to operate step 2 and 4, as source LAS and redshift are the two key factors to determine whether a source is a GRG. As a result, the Proctor (2016) candidate list has contributed more than 35% of all sources to the total confirmed GRG popularization.

In spite of its success, the algorithm presented in Proctor (2016) also raises questions about selection biases: for example, in this instance only sources with two radio components and large LAS ($\geq 4'$) are considered. Such selection biases would have excluded at least 108 GRGs of smaller LAS in the Kuźmicz et al. (2018a) catalogue. Also, it is problematic that GRGs with more complicated morphologies could not be recognized

TABLE 1.1: A summary of key parameters for each survey mentioned in this section.

Survey	Telescope	Frequency	Sky Coverage	FWHM beam	rms noise	source count
NVSS	VLA	1400 MHz	$\delta > -40°$	$45''$	0.45 mJy	~1.8 million [e]
SUMSS	MOST	843 MHz	$\delta < -30°$ [a]	$45'' \times 45'' \csc \delta$	1.3–2.5 mJy	211 063 [f]
FIRST [b]	VLA	1400 MHz	10 575 deg^2	$5''$	0.15 mJy	946 432 [g]
LoTSS[c]	LOFAR	120-168 MHz	$\delta > 0°$	$6''$	71 μJy	
EMU[d]	ASKAP	1.3 GHz	$\delta \leq +30°$	$10''$	10 μJy	
WODAN[d]	APERTIF	1400 MHz	$\delta \geq +30°$	$\sim 15''$	10 μJy	

[a] SUMSS excludes the regions covered by MGPS-2 (Murphy et al., 2007).
[b] Information on the FIRST survey was retrieved from: http://sundog.stsci.edu/first/images.html.
[c] Parameters of LoTSS in the table are recorded in accordance with LoTSS DR 1 (Shimwell et al., 2019).
[d] Survey parameters for EMU and WODAN are those expected by survey project scientists (Norris et al., 2011; Röttgering et al., 2011), which might change when the actual survey data release comes out.
[e] NVSS latest catalogue source numbers were retrieved from https://www.cv.nrao.edu/nvss/.
[f] NVSS latest catalogue source numbers were retrieved from http://www.astrop.physics.usyd.edu.au/sumsscat/news.html.
[g] FIRST latest catalogue (2014 Dec. 17^{th}) source numbers were retrieved from http://sundog.stsci.edu/first/catalogs/history.html.

by the Proctor (2016) algorithm as these are considered important for particular types of investigation. In light of these considerations, I present a GRG classifier capable of identifying GRGs of smaller LAS and with diverse radio morphologies in Chapter 5. I would use an approach based on Convolutional Neural Networks (CNN; Krizhevsky et al., 2012a). Considering the traditional approaches to GRG candidate validation using multi-frequency radio survey data, I also explore the possibility of using multi-survey image data and host galaxy redshifts as algorithm inputs. Further details will be given in Chapter 5.

1.4 Large Scale Radio Sky Continuum Surveys

In this section, I will run through the large scale radio sky continuum surveys I have worked with in this thesis (NVSS, SUMSS, FIRST) and several representative new ones performed by the SKA pathfinders (e.g. ASKAP, APERTIF, LOFAR) in accordance with Norris et al. (2013). I do not consider (a) sky polarization surveys (e.g. POSSUM Norris et al., 2013), (b) those with lower resolution such as the GaLactic and Extragalactic All-sky MWA Survey (GLEAM; Wayth et al., 2015) or (c) the deep surveys only covering a small patch of sky like the the MeerKAT International GHz Tiered Extragalactic Exploration Survey (MIGHTEE; Jarvis et al., 2016) as it might be less relevant/efficient/necessary to hunt GRGs using these survey data in an automated way. It is noteworthy that hundreds of GRGs have been found with the help of the following archival surveys (Kuźmicz et al., 2018a), and the ones conducted by SKA-pathfinder surveys I mentioned in this section are likely to find more.

1.4.1 Well-known Archival Surveys

NVSS/FIRST

The NRAO VLA Sky Survey (NVSS; Condon et al., 1998) and Faint Images of the Radio Sky at Twenty-Centimeters (FIRST; Becker et al., 1995) are two radio sky surveys that were conducted in parallel by the Very Large Array (VLA; Thompson et al., 1980). Both NVSS and FIRST have provided source catalogues with source samples of order 10^6 (around 1.8 million) via elliptical Gaussian fitting to all significant peaks (Condon et al., 1998; Becker et al., 1995).

Though the survey resolution and sensitivity of NVSS or FIRST is lower than state-of-the-art radio sky surveys like LoTSS, see Table 1.1, NVSS and FIRST data are already sufficient for astronomers to distinguish source radio morphology and identify source hosts. Since the two surveys launched, a large number of GRGs have been discovered with their help or by directly performing candidate selection from their images/catalogue data (e.g. Schoenmakers et al., 2000b; Lara et al., 2001a; Machalski et al., 2001; Schoenmakers et al., 2001; Sadler et al., 2002; Machalski et al., 2007; Solovyov & Verkhodanov, 2014; Molina et al., 2014; Amirkhanyan et al., 2015; Amirkhanyan, 2016; Dabhade et al., 2017; Clarke et al., 2017; Kapińska et al., 2017; Sebastian et al., 2018; Kozieł-Wierzbowska et al., 2020b; Dabhade et al., 2020c).

SUMSS

The Sydney University Molonglo Sky Survey (SUMSS; Bock et al., 1999; Mauch et al., 2003) is considered to be a counterpart to NVSS in the Southern hemisphere, where it covers most of the sky south of declination -30°. The survey was conducted by the Molonglo Observatory Synthesis Telescope (MOST; Mills, 1981; Robertson, 1991) in Australia, with similar angular resolution and slightly poorer survey sensitivity, see Table 1.1. Since launched, the survey has been used as the most sensitive radio sky survey in the Southern hemisphere, and has facilitated the discovery of at least 20 GRGs (Saripalli et al., 2005; Huynh et al., 2007; Dabhade et al., 2017). There are also studies using SUMSS maps for GRG candidate cross-validation (e.g. Amirkhanyan, 2016). Although the number of GRGs discovered in the Northern hemisphere is far larger than that in the Southern sky, it is claimed that the imbalanced discovery is largely caused by the lack of large scale GRG surveys in the Southern sky (Kuźmicz et al., 2018a).

1.4.2 SKA-pathfinder Surveys

LoTSS

LoTSS is a large scale radio sky survey conducted by the LOw-FRequency Array (LOFAR; van Haarlem et al., 2013) at 120-168 MHz. The survey aims to cover the whole Northern sky when finished, and had completed over 20% by the time when the LoTSS DR 1 was released (Shimwell et al., 2019). In accordance with the published LoTSS data, LoTSS will have median sensitivity of $S_{144\,\mathrm{MHz}} = 71\,\mu$Jy/beam, along with 90% completeness of

point sources at an integrated flux limit of 0.45 mJy (Shimwell et al., 2019). The angular resolution and positional accuracy of LoTSS are 6″ and within 0.2″, respectively.

Since LoTSS launched, it has been working to increase the number of catalogued young and old AGN, including GRGs (Shimwell et al., 2017). Indeed, LoTSS has several advantages that might benefit the work of GRG hunting (Williams et al., 2019; Hardcastle et al., 2019b; Duncan et al., 2019):

- (a) low operation frequency;
- (b) high potential completeness for measured source host redshifts;
- (c) a relatively well-developed pipeline to identify extended radio sources in the survey.

Since GRG lobes share steep spectral indices, they tend to be brighter at lower frequencies (Dabhade et al., 2020b). LoTSS therefore become advantageous when hunting GRGs. Also, the LoTSS team have implemented a maximum likelihood identification method to determine if a source and its counterpart in different wavelengths matched correctly. LoTSS scientists then cross matched most of their compact or mildly resolved entries with a combined PanSTARRS-WISE catalogue. For well-extended or complex entries, however, LoTSS introduced a Zooniverse citizen project LOFAR Galaxy Zoo (LGZ; Shimwell et al., 2019) to sort out the cross matching. LGZ requires project users to manually visualize maps of sample entries, correlate radio components belonging to the same source, and cross match their optical/infrared counterparts. By looking at a subset of LGZ samples and manually visualizing RGZ DR1 image data, Dabhade et al. (2020b) found 225 new GRGs, resulting in LoTSS having the highest GRG number density among existing large scale radio sky surveys.

EMU

EMU is one of the eight major sky surveys being conducted by the Australia SKA Pathfinder (ASKAP; Johnston et al., 2007, 2008). According to the survey's primary goal, EMU will achieve an angular resolution of 10″, an rms noise level of around 10 μJy/beam, operate at 1.3 GHz and cover the sky region south of declination +30°. The high sensitivity of EMU is anticipated to lead to the dense discovery of radio sources, estimated to be over 2200 sources per sq. degree (Jackson, 2005; Schinnerer et al., 2007; Scoville et al., 2007; Norris et al., 2011), of which about 75% of EMU sources would be star-forming galaxies (SFG; Seymour et al., 2008) and the rest would be AGN (Norris et al., 2011).

The challenge of EMU in terms of GRG hunting is that the survey might lack spectroscopic redshifts. Fortunately, over half of the 70 million EMU objects will receive photometric observations in 2020 using an optical/IR identification pipeline that incorporates a citizen science project (Norris et al., 2011). In particular, low-power AGN and elliptical galaxy hosted AGN at $z < 1$ are likely to have reliable photometric redshifts, and therefore it will be possible to hunt GRGs using EMU.

WODAN

The Westerbork Observations of the Deep APERTIF Northern sky survey (WODAN) is seen to be the Northern hemisphere counterpart of the ASKAP EMU survey (Röttgering et al., 2011). It is planned to observe the sky regions north of declination +30° with a surface brightness limit of 10 μJy/beam at 1400 MHz. WODAN is expected to have an angular resolution of around 15″. EMU and WODAN together would ideally cover all radio loud AGN and luminous starburst galaxies with $z \leq 2$ (Röttgering et al., 2011). WODAN objects will ideally have the spectroscopic redshifts of their host galaxies with $z \leq 0.3$ provided by SDSS. Sources of higher redshift, on the other hand, will still rely on photometric redshfit measurements. Given the high availability of spectroscopic redshifts in the Northern sky, WODAN is expected to facilitate GRG hunting at lower redshifts and achieve higher candidate confirmation fractions compared with the EMU survey, which targets the southern hemisphere.

Chapter 2

Machine Learning for Classification

2.1 Before classification: linear regression

2.1.1 Simple linear regression

It is just another day. The author walks into a hospital and wonders whether he has a fever (the author is fine but just rather nervous). The receptionist gives the author a mercurial thermometer to test his body temperature. He sees the mercury column grow as the mercury inside the thermometer expands monotonically. The test will take him several minutes, and the author starts to wonder what the exact relationship between the mercury column height and the displayed temperature will be.

Reading the ticks on the thermometer, the author believes that the mercury expansion is uniform as the temperature increases, or at least that it should be. In this case, the height of the mercury column (X) would be proportional to the patient's body temperature (Y). The relationship therefore could be summarised as:

$$\mathtt{Y} = A\mathtt{X} + B \tag{2.1}$$

where A is the slope, and B is the intercept. There are two ways to find A and B. The first is to believe the relationship is strictly linear. In this case, the author can simply learn the intercept by noting the starting point of the thermometer (e.g. 35°C), and derive its slope by examining the increments on the thermometer scale (e.g. 1.5°C). The second method is for those who question a little deeper, assuming that the relationship is linear but requiring validation. Such validation could be provided by applying simple linear regression, which is widely used in statistics. Interestingly, the invention of the mercury-in-glass thermometer by Daniel Gabriel Fahrenheit in 1714 ocurred almost a century before the official discovery of linear regression, when Gauss and Legendre independently developed the method in the 1800s (Stigler, 1986).

The method is also known as least-squares linear regression and to use this method, the author needs to make some (virtual) measurements, which can be seen in Figure 2.1[1].

[1]The code snippets in this chapter can be found on https://github.com/HongmingTang060313/ML_basics, which is inspired from *https://www.udemy.com/course/practical-deep-learning-with-pytorch/*

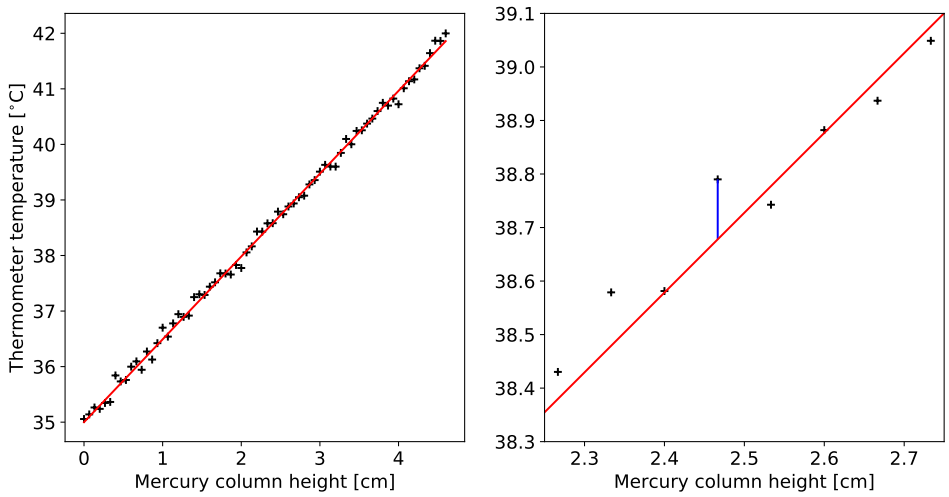


FIGURE 2.1: `Left`: An illustration of the least square linear regression method applied to examining the mercurial thermometer validity. The 70 data points on the diagram are initialized randomly based on manual measurements of a real clinical thermometer made by the author. The red line in the diagram is fitted to the data via the method of least squares. `Right`: A zoomed-in version of the left diagram, where the blue segment refers to the distance between the actual data and the fitted line.

In this figure, 70 data points are scattered around a line fitted using the least-squares method calculated using `scipy.stats.linregress`, which directly outputs the variables A and B. In this case, the author found that A is equal to 1.5, and that B is equal to 35.0, consistent with the first method. But how was this calculated?

In Figure 2.1, each measured temperature point, Y_i, can be considered to deviate from a predicted temperature (Ax_i+B) by a distance that is some random deviation value, ε_i, shown as the blue segment in Figure 2.1. The goal of the least squares method is to find a pair of A and B that minimize the sum of the squared residuals, $\sum \varepsilon_i^2$ (Stigler, 1986).

Mathematically, the goal is:

$$min\, Q(A,B) = min \sum_{i=1}^{70} \varepsilon_i^2 = min \sum_{i=1}^{70} (Y_i - A\, x_i - B)^2 \tag{2.2}$$

Describing this goal as a function, f, the function would reach its minimum when its partial derivatives reach zero. The partial derivatives are:

$$\frac{\partial f(A,B)}{\partial A} = 0 = \frac{\partial \sum_{i=1}^{70} (Y_i - A\, x_i - B)^2}{\partial A} = -\frac{2}{70} \sum_{i=1}^{70} (x_i Y_i - B\, x_i - A\, x_i^2) \tag{2.3}$$

$$\frac{\partial f(A,B)}{\partial B} = 0 = \frac{\partial \sum_{i=1}^{70} (Y_i - A\, x_i - B)^2}{\partial B} = -\frac{2}{70} \sum_{i=1}^{70} (Y_i - A\, x_i - B). \tag{2.4}$$

By substituting B as a function of A into Equation 2.4 and into Equation 2.3, we find that

$$A = \frac{70\sum_{i=1}^{70} x_i Y_i - \sum_{i=1}^{70} x_i \sum_{i=1}^{70} Y_i}{70\sum_{i=1}^{70} x_i^2 - (\sum_{i=1}^{70} x_i)^2} \tag{2.5}$$

$$B = \sum_{i=1}^{70} Y_i - A\sum_{i=1}^{70} x_i. \tag{2.6}$$

Such simple linear regression is easily accessible and feasible for manual computation. However, in many cases other than the mercurial thermometer, parameters may be dependent on more than one independent variable. The computation of such relationships requires some more complex linear algebra.

2.1.2 Multivariate linear regression

When the method of least squares involves multiple independent variables, the relationship between the independent variables, x_j, and the dependent variable, Y, becomes:

$$Y_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} \tag{2.7}$$

where β_j represents the coefficient associated with each set of variables. In this case, in order to obtain a unique solution, the number of data samples, n, has to be larger than the number of variables, k.

Similarly to simple linear regression, the goal of multiple linear regression is:

$$min\, Q(\beta_0, \beta_1, ..., \beta_k) = min \sum_{i=1}^{n} \varepsilon_i^2 = min \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij})^2 \tag{2.8}$$

Again, this goal can be achieved when the partial derivative of each independent variable becomes zero. The form is similar to that of Equation 2.3, and is written as:

$$\frac{\partial f(\beta_0, \beta_1, ..., \beta_k)}{\partial \beta_0} = 0 = \frac{\partial \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij})^2}{\partial \beta_0} = \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij}) \tag{2.9}$$

$$\frac{\partial f(\beta_0, \beta_1, ..., \beta_k)}{\partial \beta_j} = 0 = \frac{\partial \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij})^2}{\partial \beta_j} = \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij}) x_{ij};\ j = 1, 2, ..., k \tag{2.10}$$

By solving this set of equations, one can obtain a set of coefficients if a unique solution exists. However, given that such relationships are more easily formulated as vectors and

matrices, it is preferable to use them in this form:

$$\mathbf{Y} = [y_1, y_2, ..., y_n]^T \tag{2.11}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & ... & x_{1k} \\ 1 & x_{21} & x_{22} & ... & x_{2k} \\ 1 & ... & ... & ... & x_{ik} \\ 1 & x_{n1} & x_{n2} & ... & x_{nk} \end{bmatrix} \tag{2.12}$$

$$\beta = [\beta_1, \beta_2, ..., \beta_k]^T \tag{2.13}$$

$$\varepsilon = [\varepsilon_1, \varepsilon_2, ..., \varepsilon_n]^T \tag{2.14}$$

In this case, Equation 2.7 becomes:

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon, \tag{2.15}$$

and the total squared error is simplified as:

$$\sum_{i=1}^{n} \varepsilon_i^2 = (\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta). \tag{2.16}$$

If the total squared error ε^2 is minimized - ideally equal to zero - the fitted values $\hat{\mathbf{Y}}$ could be represented as:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta}. \tag{2.17}$$

In an n-dimensional space, the column space of $\mathbf{X}$ can be seen as the $n-1$ dimension plane and $\hat{\mathbf{Y}}$ would become the projection of $\mathbf{Y}$. $\mathbf{Y}$-$\hat{\mathbf{Y}}$ should then be orthogonal to the plane. When $\hat{\beta}$ satisfies Equation 2.16, the derivatives give:

$$0 = \mathbf{X}^T((\mathbf{Y} - \mathbf{X}\hat{\beta}) = \mathbf{X}^T\mathbf{Y} - \mathbf{X}^T\mathbf{X}\hat{\beta} \tag{2.18}$$

By solving this equation, one can obtain one or more sets of $\hat{\beta}$ as solutions. When the inverse of $\mathbf{X}^{'}\mathbf{X}$ exists, or say the columns of $\mathbf{X}$ are independent of each other, $\hat{\beta}$ will have a unique solution:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{2.19}$$

Sounds great! However, in the real world, one might find it difficult to keep all the $\mathbf{X}$ columns independent, as the problems we meet daily are usually not well-understood. When facing such challenges, rather than finding a global minimum by analytic computation, it seems more practical to gradually decrease the least squared error using numerical methods and this is the basis of many popular machine learning approaches.

In [1]:
```
import numpy as np
import torch
from torch.autograd import Variable
import matplotlib.pyplot as plt
```

In [2]:
```
tem = np.arange(start=35,stop=42,step=0.1)
col = [(i-35)/1.5 for i in tem]
noise = np.random.normal(0,0.1,len(col))
new_tem = []
for ii in range(len(col)):
    new_tem.append(tem[ii]+noise[ii])
```

In [3]:
```
# arrange data for training
x_train = np.array(col, dtype=np.float32)
x_train = x_train.reshape(-1, 1)
print("x train array shape:",np.shape(x_train))
y_train = np.array(new_tem, dtype=np.float32)
y_train = y_train.reshape(-1, 1)
print("y train array shape:",np.shape(y_train))
```

```
x train array shape: (70, 1)
y train array shape: (70, 1)
```

In [4]:
```
class linearRegression(torch.nn.Module):
    def __init__(self, inputSize, outputSize):
        super(linearRegression, self).__init__()
        self.linear = torch.nn.Linear(inputSize, outputSize)

    def forward(self, x):
        out = self.linear(x)
        return out
```

FIGURE 2.2: Example code showing the data sample and the simple linear regression model foundations used to solve the thermometer temperature prediction problem in Section 2.1.1. The model class module is built with the PYTHON `Pytorch` package.

2.1.3 Machine learning for regression: the mercurial thermometer example

Rather than taking an analytic approach to solving a simple linear regression problem, supervised machine learning aims to find an iterative numerical solution to the problem. In this context, 'supervised' learning methods generally maps the inputs to a output using a training data set with known (input,output) correspondence (Russell & Norvig, 2009). Such models will initialize a model with randomly valued parameters, calculate the temporary sum of squared residuals, and then gradually update those parameters under some given rules. The goal of such an approach is the same as that of the analytic one, which is to minimize the sum of squared residuals. In this section I will run through a simple, step-by-step machine learning approach to solving the mercurial thermometer problem. I will use the `python Pytorch` package in this example, and the `jupyter` notebook open source web application to run and visualize the code.

2.1.4 Initialization

To train a machine learning algorithm, the first step is to prepare a training data set. As shown in Cell 2 of Figure 2.2, here 70 pairs of mercury column height and virtually measured temperature data are initialized. The noise associated with each virtual measurement is set to have a standard deviation of 0.1°C. The data for mercury column height, X, and temperature, Y, are then converted into the `Variable` format for model training in `Pytorch`. It can be seen that both X and Y have identical dimensions, and these data are now ready for machine learning training.

The second step of training is to define the model structure. In `Pytorch`, this can be done by defining a model 'class'. By specifying the `inputSize` and `outputSize` to be 1,

```
In [6]: criterion = torch.nn.MSELoss()
        optimizer = torch.optim.SGD(model.parameters(), lr=learningRate)

In [7]: for epoch in range(epochs):
            # inputs/labels to variables fitted for Pytorch
            inputs = Variable(torch.from_numpy(x_train))
            labels = Variable(torch.from_numpy(y_train))

            # Clear gradient buffers from previous epochs
            optimizer.zero_grad()

            # model output with given inputs
            outputs = model(inputs)

            # predicted output loss
            loss = criterion(outputs, labels)

            # get gradients w.r.t to parameters
            loss.backward()

            # update parameters
            optimizer.step()
```

FIGURE 2.3: Example code showing the process to train the model defined in Figure 2.2. Consistent with the description in Section 2.1.4, I define the model loss function to be MSE and use SGD as the optmization method in Cell 6. The iterative model training process is defined in Cell 7.

and the output to be a linear combination of inputs, a raw simple linear regression model is defined. The model parameters, here the weight, w, and bias, b, of the `Pytorch.nn` linear, i.e. fully-connected, layer are initialized as well.

The third step is to define a loss function and optimization method for the training process. For regression problems, the most commonly used loss function is the Mean Square Error (MSE) loss function (Paszke et al., 2019), which computes the mean of the squared error between the prediction and the training data.

The optimization method, on the other hand, is the machine learning strategy one adopts. In this example, we apply the Stochastic Gradient Descent (SGD) method, usually regarded as a stochastic approximation of the gradient descent method (Robbins & Monro, 1951). When training on input data, the model will stochastically initialize the model parameter with small values, compute the gradient of the loss function, and then decrease the loss at a rate proportional to the corresponding gradient. This process is explained in more detail in the following section. For this example, the model parameters are updated after each training *epoch*: all training data are passed once through the model per epoch, see Figure 2.3. The learning rate under the SGD regime is a customized update step rate scalar that links the loss gradient and the existing parameter values. The larger the learning rate, the more aggressively the model will update the parameters. In this example, the learning rate is defined to be 0.1.

The training process of the machine learning algorithm iteratively minimizes the MSE loss, which is visualized in Figure 2.4. Over 150 epochs of training, the model decreases its MSE loss from a few hundred to a value of only 0.009. Considering the nature of the least squared loss, this means that the average offset between each predicted temperature and actual temperature measurement is less than 0.1°C. A more direct visualization can be seen in Figure 2.5, where the weight, w, and bias, b, of the fitted model gradually

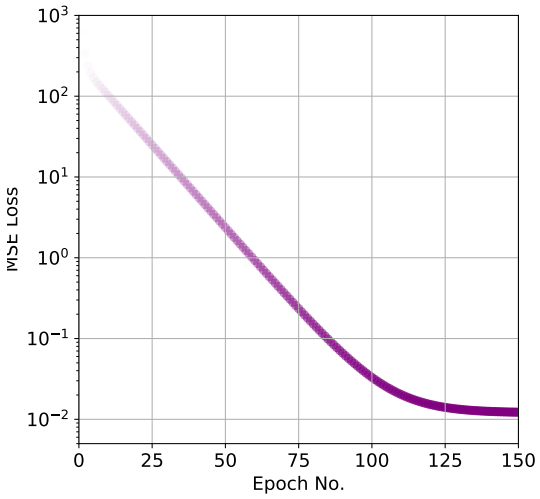


FIGURE 2.4: An illustration of the learning loss curve for the temperature prediction problem, when the optimization aims to minimize the model MSE loss.

grows from its randomly selected initial values to the expected analytic solution. The learning process is smooth, mostly due to the fact that this question has an analytic solution, allowing the model loss to have a single minimum, as well as the fact that the noise level of the input data is low.

2.1.5 Back-propagation

As shown in the previous section, the machine learning algorithm is trained by repeatedly passing the input training data forward through the model, computing the MSE loss gradient with respect to the model parameters at every training epoch, and then decreasing the MSE loss by optimizing the parameters at a rate proportional to the loss gradient. This process of parameter optimization is referred to as *back-propagation* in the context of artificial neural network models (Goodfellow et al., 2016a).

For the simple SGD optimization example shown above, mathematically each parameter update looks like:

$$P_{new} = P_{old} - learning\,Rate \times G_{new} \tag{2.20}$$

where P_{old} are the existing model parameters, and P_{new} are the updated parameters after an epoch of training; G_{new} refers to the loss gradient as a function of parameter, as given by Equations 2.3 & 2.4 for this example.

This is illustrated in Figure 2.6 where the gradient of the loss for each parameter in the `Pytorch` model is shown and compared with gradients computed using the analytic expressions. It can be seen that the numerical gradients are highly consistent with the analytic ones, which in principle allows one to manually train a simple linear regression model.

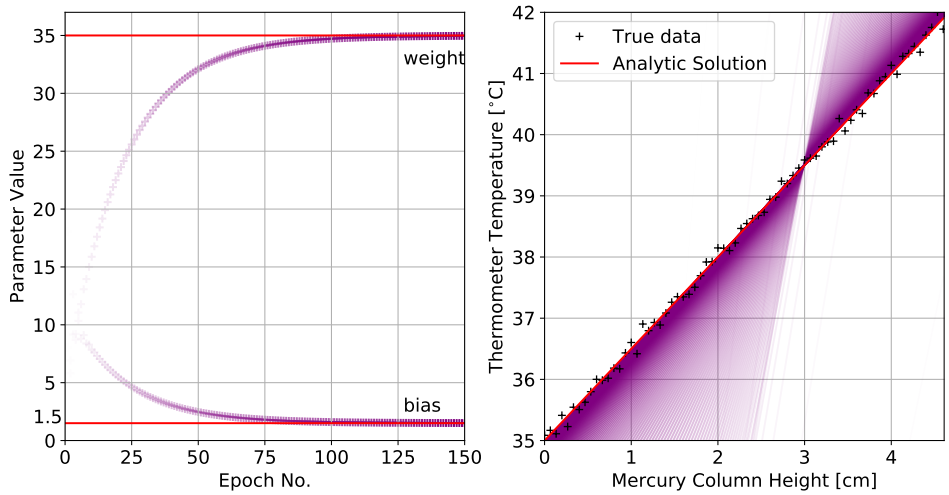


FIGURE 2.5: Left: An illustration of the model weight/bias parameter optimization process during the simple linear regression model training process. Red horizontal lines show the analytical solutions of the problem. Right: The resulting predicted linear regression formula after each epoch of model training. Black crosses show the true data points considered in the problem, and the red solid line shows the analytical solution.

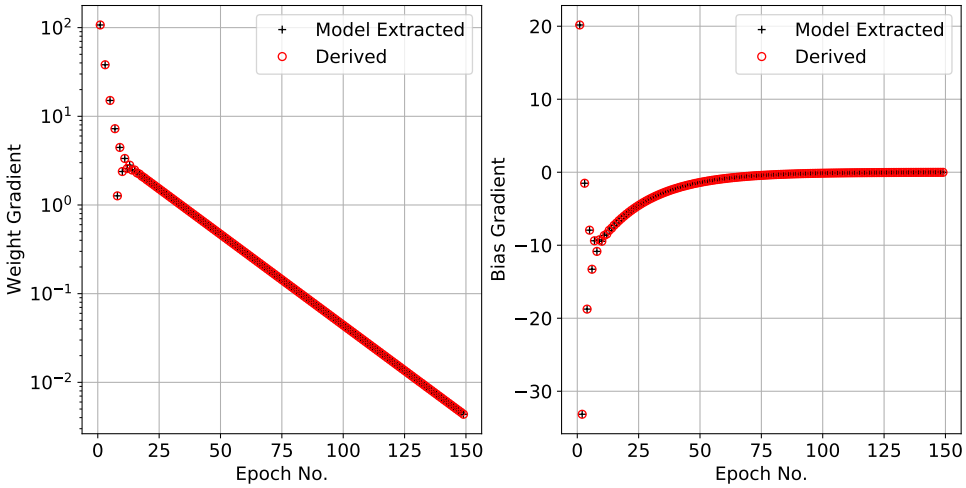


FIGURE 2.6: Left: An illustration of the temperature prediction model weight gradients during the 150-epoch model training, where the + signs and the red circles represent the weight gradients extracted from the model directly and those derived mathematically, respectively. Right: The same diagram for the model bias parameter.

2.2 Migration to Logistic Regression

Looking back at the above example of the mercurial thermometer, we have trained a simple *linear regression* model using machine learning to predict a temperature value based on the height of mercury in the thermometer. In comparison, *logistic regression* algorithms do not result in direct temperature estimation but instead can provide probabilistic predictions to similar questions: for example, does the author have a fever or not?

In this section, we will run through the fever detection problem. This problem requires the same mercurial thermometer, and is solved by training a logistic regression algorithm to detect fever. The key to the problem is to have the algorithm recognize the assumed temperature limit for diagnosing a fever: 37.1°C. In other words, the fitting function has to make a non-linear prediction.

2.2.1 From linear to non-linear

The goal of such a model is to return a high probability (e.g. $p \to 1.0$) of fever when one's body temperature is measured to be higher than 37.1°C, and to ensure that the model would output a low probability (e.g. $p \to 0.0$) when one's body temperature is below this limit. An example of such a representative function for achieving this goal is the Sigmoid function (Mira & Hernández, 1995). Mathematically, the Sigmoid function has the following format:

$$q = \frac{1}{1 + e^{-X}}, \tag{2.21}$$

where X is the input temperature, and q is the predictive probability of having a fever. In the context of machine learning, the standard form of the Sigmoid function is untrainable, as it would simply normalize any given input value to a value between 0 and 1. In practice, machine learning approaches (e.g. the `Pytorch` setup) adopt the following form of the Sigmoid function:

$$\mathtt{q} = \frac{1}{1 + e^{-Y}} = \frac{1}{1 + e^{-(w\mathtt{X}+b)}} \tag{2.22}$$

where w and b are the 'slope' and 'bias' in Equation 2.1, or 'weight' and 'bias' in the context of machine learning. These two parameters can be optimized in a step-wise manner similar to that shown in the last section.

In the context of logistic regression we define the input X to be the difference between the temperature data point and the fever limit: 37.1°C. We further define Y based on the value of X: if an input is positively valued, its corresponding output is unity, otherwise the output will have a value of zero. These inputs and newly defined outputs are the training data that will be used in this section.

2.2.2 Cross Entropy Loss

Since the intended output of the model is a probability, we here adopt a loss function that is appropriate for probability prediction: the cross entropy loss (Goodfellow et al.,

2016a). Cross entropy is a measure of similarity between two probability distributions, here denoted p and q, where q is the predicted probability distribution, while p is the ground truth probability distribution. The smaller the cross entropy, the closer the two distributions are considered to be. The binary cross entropy loss is defined as:

$$H(p,q) = -\frac{1}{N}\sum_{i=1}^{N}(p_i \log(q_i) + (1-p_i)\log(1-q_i)). \tag{2.23}$$

By substituting Equation 2.22 into Equation 2.23, the loss gradients for the mercury thermometer problem with respect to the model weight and bias at each epoch are:

$$\frac{\partial H}{\partial w} = \frac{\partial H}{\partial q}\frac{\partial q}{\partial Y}\frac{\partial Y}{\partial w} = -\frac{1}{70}\sum_{i=1}^{70}(\frac{p_i}{q_i} - \frac{1-p_i}{1-q_i}) \times \frac{e^{-Y}}{(1+e^{-Y})^2} \times x_i, \tag{2.24}$$

$$\frac{\partial H}{\partial b} = \frac{\partial H}{\partial q}\frac{\partial q}{\partial Y}\frac{\partial Y}{\partial b} = -\frac{1}{70}\sum_{i=1}^{70}(\frac{p_i}{q_i} - \frac{1-p_i}{1-q_i}) \times \frac{e^{-Y_i}}{(1+e^{-Y_i})^2} \times 1, \tag{2.25}$$

and these loss gradient functions provide the value of G_{new} in Equation 2.20.

2.2.3 Theory vs. Reality

The model setup for the logistic regression problem is similar to that of linear regression. Two hyper-parameters that are different are the model learning rate and the number of training epochs, which have been set to 0.5 and 100, respectively. The other change is that we add the `torch.sigmoid` function to the model setup, ensuring the model output follows a non-linear Sigmoid distribution.

The training loss and parameter convergence are shown in Figures 2.8 & 2.7. As the cross entropy loss decreases, the model parameters are gradually optimized to approximate the intended probability distribution. This process is also seen in Figure 2.9, where the loss gradient with respect to the model weight and bias moves towards zero. Figure 2.9 also compares the parameter loss gradients extracted from the `Pytorch` model with the analytic gradients computed by hand. It can be seen that these are equivalent. In other words, equipped with the aforementioned functions, one could build and train the model by hand; however, solving such a simple problem is just the starting point for developing logistic regression machine learning algorithms. In the following sections, I will start to explore feed-forward neural networks (Goodfellow et al., 2016a) in more detail, where additional non-linearities are introduced and the machine learning approach becomes more powerful.

2.3 Feedforward Neural Networks 1: Basics

Feedforward Neural Networks were the first and simplest type of Artificial Neural Network (ANN) to be produced and implemented in astronomical researches (e.g., Adorf & Meurs, 1988; Storrie-Lombardi et al., 1992; Lahav et al., 1995). Such networks have only

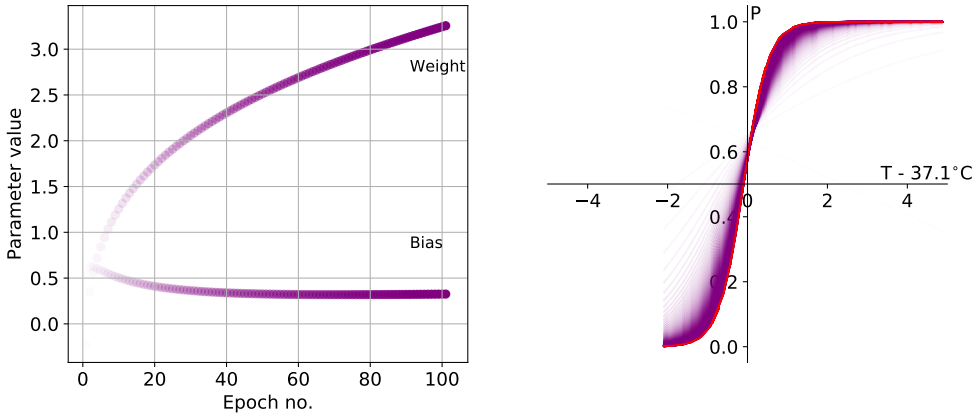


FIGURE 2.7: Left: An illustration of the simple logistic regression model parameter evolution during the training process. Right: An illustration of the fitted Sigmoid function as a function of training epoch. The opacity of the lines increases as a function of training epoch. T and P in the diagram refer to the data input temperatures and output probabilities. The red curve represents the final Sigmoid function when the model is finished training.

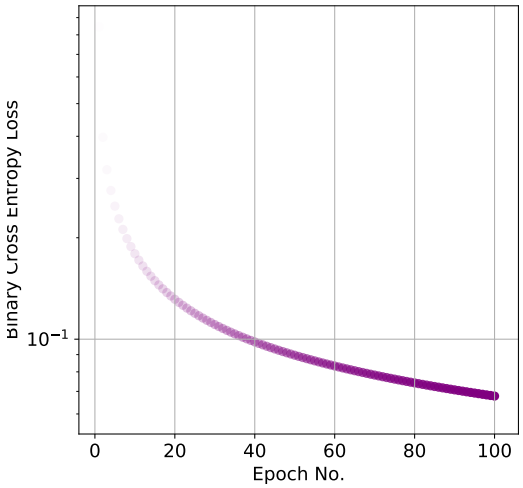


FIGURE 2.8: The training loss curve for the fever alarm model, where we adopt the Binary Cross Entropy loss function. The opacity of the data points increases as a function of training epoch.

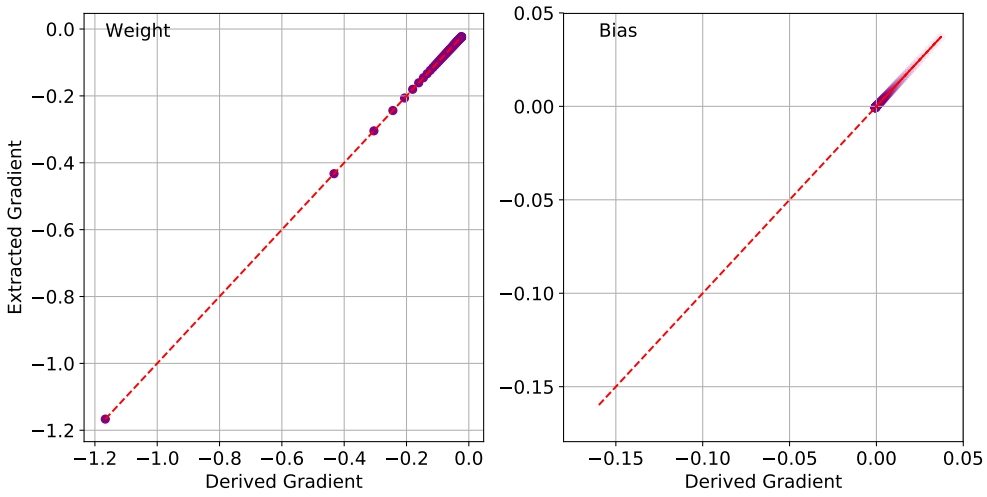


FIGURE 2.9: Derived parameter gradients vs. Extracted model parameter gradients. The red dashed line in both subplots represent the 'equal line', where derived parameter gradients are consistent to the extracted ones. Color transparency of each data point is proportional to its training epoch number.

a forward pass, with no loop or cycle. The logistic regression model presented in the last section is actually an example of the simplest one-neuron feed-forward neural network. However, before introducing further details of FNNs, there is a question to be addressed: what does the term *neuron* mean in the context of artificial neural networks?

2.3.1 The Artificial Neuron

The concept of an artificial neuron is based on biological neurons, and is the elementary building block in Artificial Neural Networks. Each neuron receives one or more inputs, sums those inputs and produces a single output. In a typical case, each input will be weighted independently, and the output will be passed through a transfer or *activation* function. Mathematically:

$$y = \phi(\sum_{i=1}^{k} w_i x_i + b_i), \tag{2.26}$$

where we assume that the neuron has k input variables, x_i; ϕ is the transfer function; and y in this equation represents the neuron output. Thinking of the linear regression problem, our example was a single neuron neural network without a transfer function. On the other hand, the logistic regression example introduced the Sigmoid function as the transfer function. In both examples, k in Equation 2.26 was equal to one. When facing multi-variate linear regression problems, k would be greater than 1 and depend on the given number of input variables.

Since we are now aware of the foundation of artificial neurons, it's time to build a slightly more complex feedforward neural network with 10 neurons. This FNN is similar to the one introduced in 1998 by Yann LeCun, Corinna Cortes, and Christopher J.C.

Burges (Lecun et al., 1998b), who were trying to solve the problem of recognizing numbers from images of hand written digits - a significantly more complex problem than the previous fever alarm example.

2.3.2 MNIST: identifying hand written digits

Similarly to the previous sections, the first step in training a machine learning model is to prepare a well-labelled training data set. Rather than creating the samples ourselves this time, we here adopt a well-known image classification database: the Modified National Institute of Standards and Technology database (MNIST; Lecun et al., 1998b). The MNIST database contains 60,000 greyscale training sample images and a further 10,000 samples for testing. Each image in the database shows a hand-written digit (0 to 9) in a field of 28 $\times$ 28 pixels. Since the MNIST database launched, it has been widely adopted in image processing systems (e.g. Lecun et al., 1998b; Keysers et al., 2007; Hasanpour et al., 2016; Kowsari et al., 2018), and for the development of machine learning algorithms.

The MNIST database is a re-mixed subset of the larger NIST special dataset 1 (SD1) and 3 (SD3) (Lecun et al., 1998b). When the NIST datasets were founded, their training set (SD3) was produced by American office workers, while the test set (SD1) was created by American high school students. The differing writer constitution between the two datasets makes the data samples in SD3 much cleaner and easier to recognize than those of SD1 (Lecun et al., 1998b) and the original NIST dataset is therefore considered less suitable for developing machine learning algorithms. In order to overcome the NIST SD3/SD1 short-comings, the MNIST database picked half of its training/testing samples from the NIST training set and the other half from the NIST test set, ensuring that the sets of writers are distributed evenly between the two datasets. MNIST futher normalized and centered the NIST images, so that each one would fit within the 28 $\times$ 28 pixel boundaries (Lecun et al., 1998b).

As already mentioned, the MNIST database has been adopted into a variety of machine learning algorithm development, including FNNs and Convolutional Neural Networks (CNNs), which I will introduce in the next section. I therefore adopt MNIST as the example data set for the remainder of this chapter.

2.3.3 10-neuron FNN: Theory

When the MNIST dataset was first published, a simple 10-neuron feedforward neural network was proposed along with its release (Lecun et al., 1998b). The number of ten neurons was selected because the MNIST database was being used to classify hand written digits from 0 to 9. In the context of probability distributions, a sample image with a hand written '9' on it would have its predicted probability defined in one-hot vector form as [0,0,0,0,0,0,0,0,0,1]. Since each neuron can only have a single output, the simplest FNN approach to describe the 10-class problem therefore required 10 outputs to provide a vector fully sampling each image's probability distribution.

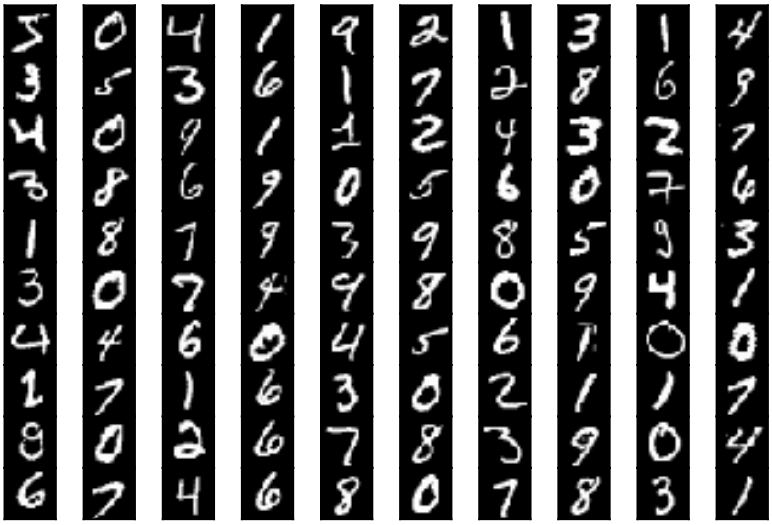


FIGURE 2.10: An sample illustration of the MNIST dataset. The diagram is showing hand written digit samples in the 10 by 10 manner. Each sample image has a size of 28 $\times$ 28 pixels, with pixel values normalized from 0 to 1.

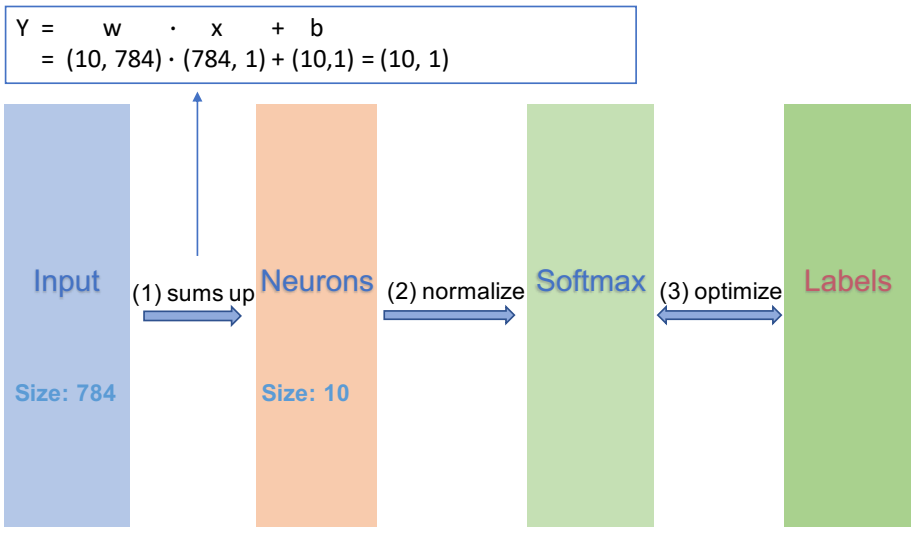


FIGURE 2.11: A schematic diagram of the 10-neuron FNN architecture. The double-sided arrow on the diagram refers to both model outputs forwarding and model parameter optimization via back-propagation.

In practice, each pixel of an image would be imported to every neuron to form a single layer 10-neuron FNN like that shown in Figure 2.11. It can be seen from the figure that the network has 7850 model parameters, with 7840 model weight parameters and 10 model bias parameters. Generally speaking, each iteration of model training can be split into three steps:

1. Each image is reshaped into a one-dimensional matrix with a size of (28×28, 1) and combined with the model weights via a dot product multiplication. The resulting output is then summed with the matrix of biases and passed to the transfer function.

2. The output is passed through the transfer function and normalized as a 10-element vector, ensuring that the sum of the vector elements is unity.

3. The resulting vector from each image input is then seen as a probability distribution, q, with its true label as p. The network uses these vectors to compute the cross entropy of the two distributions and calculate the loss gradients with respect to the model parameters.

For Step 2 we adopt the Softmax function (Goodfellow et al., 2016b) as the transfer function for this example. The Softmax function has the form:

$$\sigma(x_i)_j = \frac{e^{-\beta_j x_i}}{\sum_j^k e^{-\beta_j x_i}}, \tag{2.27}$$

where x_i is an input, and j refers to the j^{th} target class for the specific problem. The Softmax function can be seen as an extension of the Sigmoid function that we used in previous sections, as:

$$\sigma(x_i)_0 = \frac{e^{-\beta_0 x_i}}{e^{-\beta_0 x_i} + e^{-\beta_1 x_i}} = \frac{e^{-(\beta_0-\beta_1)x_i}}{e^{-(\beta_0-\beta_1)x_i} + 1} = \frac{e^{-\beta x_i}}{1 + e^{-\beta x_i}} = 1 - sigmoid(x_i)_1, \tag{2.28}$$

$$\sigma(x_i)_1 = \frac{e^{-\beta_1 x_i}}{e^{-\beta_0 x_i} + e^{-\beta_1 x_i}} = \frac{1}{1 + e^{-(\beta_0-\beta_1)x_i}} = \frac{1}{1 + e^{-\beta x_i}} = sigmoid(x_i)_1, \tag{2.29}$$

where $\beta = \beta_0 - \beta_1$. Compared to the Sigmoid function, the advantage of using a Softmax function is that it can provide a probability distribution where all the individual class probabilities add up to 1, while the Sigmoid function will assign a probability to each class between 0 and 1. For a mutually exclusive class identification problem such as MNIST a model that adopts the Softmax function is more likely to avoid predicting equal probabilities for multiple classes at the same time, thus giving a more confident class prediction.

2.3.4 10-neuron FNN: Back propagation

The back-propagation process for the 10-neuron FNN is similar to that of the logistic regression model. However, given that the MNIST model now has multiple inputs and

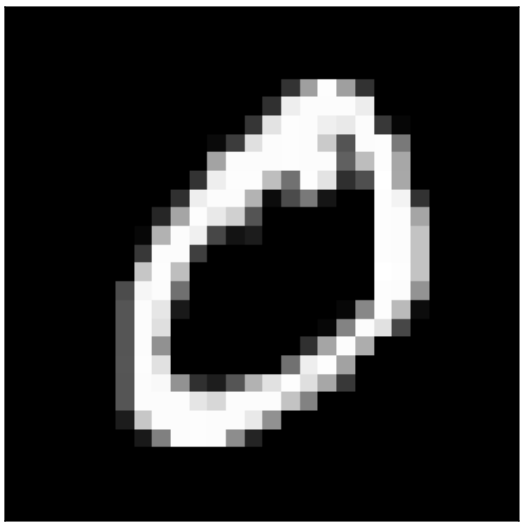


FIGURE 2.12: A MNIST sample image of hand written digit 0. The image is in grayscale, with pixel size of 28×28.

outputs, we here re-frame the problem using linear algebra. To start with, we focus on a single MNIST sample image of the hand written digit 0, as shown in Figure 2.12. As mentioned previously, the image input will be reshaped to a 1-dimensional vector, with length of $28 \times 28 = 784$. The vector can be represented as:

$$\mathbf{x} = [x_1, x_2, ..., x_{784}]^T \tag{2.30}$$

Now, imagine you are an arbitrary element in the image input vector and are being passed through the model. Given that the model has 10 neurons that are all importing the same image at the same time, you have to clone nine of yourself. Afterwards, you enter one of the neurons while your clones go to the other neurons. The neuron you enter will give you a unique weight to multiply with and a uniform value (bias) to add. This can be represented as:

$$\mathbf{Y} = \left(\begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} & \cdots & w_{0,784} \\ w_{1,0} & w_{1,1} & w_{1,2} & \cdots & w_{1,784} \\ w_{2,0} & \cdots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ w_{9,0} & w_{9,1} & w_{9,2} & \cdots & w_{9,784} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_{784} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ \vdots \\ b_9 \end{bmatrix} \right) = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_9 \end{bmatrix}. \tag{2.31}$$

The output, Y, of the 10 neurons will then be passed through the Softmax function:

$$\mathbf{S} = \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ \vdots \\ s_9 \end{bmatrix} = \mathrm{S}(\mathbf{Y}) = \begin{bmatrix} \frac{e^{y_0}}{\Sigma} \\ \frac{e^{y_1}}{\Sigma} \\ \vdots \\ \vdots \\ \frac{e^{y_9}}{\Sigma} \end{bmatrix}, \text{where} \, \Sigma = \sum_{i=0}^{9} e^{y_j}. \tag{2.32}$$

Finally, the normalized output, S, will be combined with the true image class probability distribution, p, to calculate the cross entropy loss:

$$H(\mathbf{P}, \mathbf{S}) = -\sum_{i=0}^{9} p_i \log s_i = -\log s_0 \;\; \text{where} \;\; p_0 = 1. \tag{2.33}$$

It is noteworthy that the cross entropy loss here can be simplified as the negative log of s_0, since the image has a true class probability distribution $\mathbf{P} = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. Such a simplification has a huge impact on the computational cost of calculating loss gradients.

The computation of the parameter-wise loss gradients requires us to calculate partial derivatives using the chain rule, similarly to Equation 2.24 and 2.25. Given the nature of backpropagation, we start by deriving $\partial\mathbf{H}/\partial\mathbf{S}$. For the example image input we used this is easy to derive:

$$(\frac{\partial \mathbf{H}}{\partial \mathbf{S}})^T = \begin{bmatrix} -\frac{1}{s_0} \\ 0 \\ \\ \vdots \\ 0 \end{bmatrix}, \quad \text{where} \, p_0 = 1, \tag{2.34}$$

and such a result will be similar when importing other images. The only difference will be that the index 0 needs to be exchanged to k, where p_k is equal to 1.

Secondly, $\partial\mathbf{S}/\partial\mathbf{Y}$: given that the Softmax function has contributions from all of the input parameters, the partial derivatives can be written as

$$\frac{\partial s_i}{\partial y_j} = \frac{\partial \frac{e^{y_i}}{\Sigma}}{\partial y_j} = \frac{e^{y_i}\,\Sigma - e^{y_j}\,e^{y_i}}{\Sigma^2} = s_i - s_i\,s_j = s_i - s_i^2; \;\; if \;\; i = j. \tag{2.35}$$

In the case of $i \neq j$, the first term would become 0. The complete derivatives can be shown in the matrix form as:

$$\frac{\partial \mathbf{S}}{\partial \mathbf{Y}} = \begin{bmatrix} s_0 - s_0^2 & -s_0\,s_1 & -s_0\,s_2 & \cdots & -s_0\,s_9 \\ -s_1\,s_0 & s_1 - s_1^2 & -s_1\,s_2 & \cdots & -s_1\,s_9 \\ -s_2\,s_0 & \cdots & s_2 - s_2^2 & \cdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ -s_9\,s_0 & -s_9\,s_1 & -s_9\,s_2 & \cdots & s_9 - s_9^2 \end{bmatrix}. \tag{2.36}$$

Finally, $\partial \mathbf{Y}/\partial \mathbf{W}$ and $\partial \mathbf{Y}/\partial \mathbf{B}$: by looking at Equation 2.31, we can find that the partial derivatives of model parameters are:

$$\frac{\partial y_i}{\partial w_{t,j}} = x_j \quad if \quad i = t, \tag{2.37}$$

$$\frac{\partial y_i}{\partial b_t} = 1 \quad if \quad i = t. \tag{2.38}$$

In matrix form, $\partial \mathbf{Y}/\partial \mathbf{W}$ can be represented as a matrix of size $10 \times (784 \times 10)$,

$$\begin{bmatrix} x_0 & x_1 & \cdots & x_{783} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & x_0 & x_1 & \cdots & x_{783} \end{bmatrix} \tag{2.39}$$

and $\partial \mathbf{Y}/\partial \mathbf{B}$ as an diagonal matrix of size 10×10:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}. \tag{2.40}$$

Together these matrices can be used to derive the parameter gradient with respect to the cross entropy loss. By multiplying $\partial \mathbf{H}/\partial \mathbf{S}$ and $\partial \mathbf{S}/\partial \mathbf{Y}$, we obtain

$$(\frac{\partial \mathbf{H}}{\partial \mathbf{S}} \cdot \frac{\partial \mathbf{S}}{\partial \mathbf{Y}})^T = \begin{bmatrix} 1 - s_0 \\ -s_1 \\ -s_2 \\ \vdots \\ -s_9 \end{bmatrix}, \tag{2.41}$$

and the loss gradients with respect to the model weights become:

$$\frac{\partial \mathbf{H}}{\partial \mathbf{S}} \cdot \frac{\partial \mathbf{S}}{\partial \mathbf{Y}} \cdot \frac{\partial \mathbf{Y}}{\partial \mathbf{W}} = [(x1 - s_0)x_0, \cdots (x1 - s_0)x_{783}, -s_1 x_0, -s_1 x_1, \cdots - s_1 x_{783}, \cdots - s_9 x_{783}]. \tag{2.42}$$

Considering the form of $\partial \mathbf{Y}/\partial \mathbf{B}$, the full loss gradient with respect to the model bias will be identical to $\frac{\partial \mathbf{H}}{\partial \mathbf{S}} \cdot \frac{\partial \mathbf{S}}{\partial \mathbf{Y}}$. The 10-neuron FNN is therefore able to update the parameters and optimize the digit class prediction performance. In the following section I will run through model training and explain the technical differences between practical model training using `Pytorch` and the theoretical training workflow.

```
import torch
import torch.nn as nn
import torchvision.transforms as transforms
import torchvision.datasets as dsets
```

```
train_dataset = dsets.MNIST(root='./data',
                            train=True,
                            transform=transforms.ToTensor(),
                            download=True)
```

```
batch_size = 100
```

```
train_loader = torch.utils.data.DataLoader(dataset=train_dataset,
                                           batch_size=batch_size,
                                           shuffle=True)
```

FIGURE 2.13: Example PYTHON code to load the MNIST training dataset via `Pytorch`. Image data would be loaded using a batch size of 100 (100 sample images every time).

10-neuron FNN: Training with Pytorch

Now I have introduced the training data set and the back-propagation mechanism, we can practically train the model. Similarly to the previous examples, I adopt the same setup as in Figure 2.2, now with `inputSize` equal to 784 and `outputSize` equal to 10. I also use the SGD optimizer, along with a smaller model learning rate of 0.0001. A new `torch` function that I adopt here is `torch.nn.CrossEntropyLoss`, the loss function to compute the model cross entropy loss. The function is different to the binary cross entropy function in the last example, as it combines the Softmax normalization function with the Cross Entropy loss function. Mathematically, the function is described as:

$$loss(y, class) = -\log(\frac{e^{y_{class}}}{\sum_j e^{y_i}}) = -y_{class} + \log(\sum_j e^{y_i}). \tag{2.43}$$

This combined design helps to minimise computation when a model is predicting an image class. The image class will be directly predicted without the involvement of a separate normalization function, as the predicted image class is determined by the largest output value from the 10 neurons.

Now back to the training. In a similar manner to the previous examples, we firstly import the MNIST training set to the script, as shown in Figure 2.13. The second step is then to define the training strategy. Since the MNIST database is far larger than the data samples I used in the thermometer examples, it is necessary to introduce two new hyper-parameters: *batch* and *iterations*. These two, along with *epoch*, can be explained as follow:

- **batch**: the number of samples imported to model training at each *iteration*.
- **iterations**: the number of single sample batch forward/backward processes.
- **epochs**: the number of times that the full data set is used during training.

```python
for epoch in range(num_epochs):
    for i, (images, labels) in enumerate(train_loader):
        # Load images as Variable
        images = images.view(-1, 28*28).requires_grad_()
        labels = labels

        # Clear gradients w.r.t. parameters
        optimizer.zero_grad()

        # Forward pass to get output/logits
        outputs = model(images)

        # Calculate Loss: softmax --> cross entropy loss
        loss = criterion(outputs, labels)

        # Getting gradients w.r.t. parameters
        loss.backward()

        # Updating parameters
        optimizer.step()
```

FIGURE 2.14: The example code we used to train the 10-neuron FNN via PYTHON Pytorch package

Given that the MNIST training set has 60 000 samples, the relationship between the three hyper-parameters can be expressed as:

$$\text{epochs} = \frac{\text{iterations}}{\left(\frac{60\,000}{\text{batch}}\right)}. \tag{2.44}$$

Such a relationship can be used to interpret the thermometer logistic regression problem as well. In that case, the model was trained using a batch size consistent with the training set size, and thus the model has an iteration number equal to the number of training epochs. In this MNIST example on the other hand, we define the number of iterations to be 30 000, which corresponds to processing 50 epochs of training data.

Finally, it's time for the actual training. Figure 2.14 shows the necessary `python` steps to train the model. The loops specify that on every iteration the model loads 100 training images and their labels. Within the loop, the script reforms the shape of input images to one dimension. After that, the gradients (if any) with respect to the model parameters in the previous iteration are cleared. The commands that then follow are the same in Cell 7 of Figure 2.3: forward the data to the model, compute the loss, derive the gradients with respect to the model parameters and update the value of the model parameters according to those gradients and the customized model learning rate.

It is noteworthy that `iteration` does not appear explicitly within the training setup. However, the number of iterations that the model has processed can serve as a 'magic number' to track model training within an individual training epoch. In this example, we require the model to make a prediction and compute the training loss based on 100 arbitrary images/labels every 500 iterations.

Similarly to previous examples, I show the loss gradient with respect to the model weight parameters in Figure 2.15. It can be seen that the model weights are gradually converging to zero, and still have space to learn. This can also be seen from Figure 2.16, where the training loss continues to decrease smoothly and is still far from stationary.

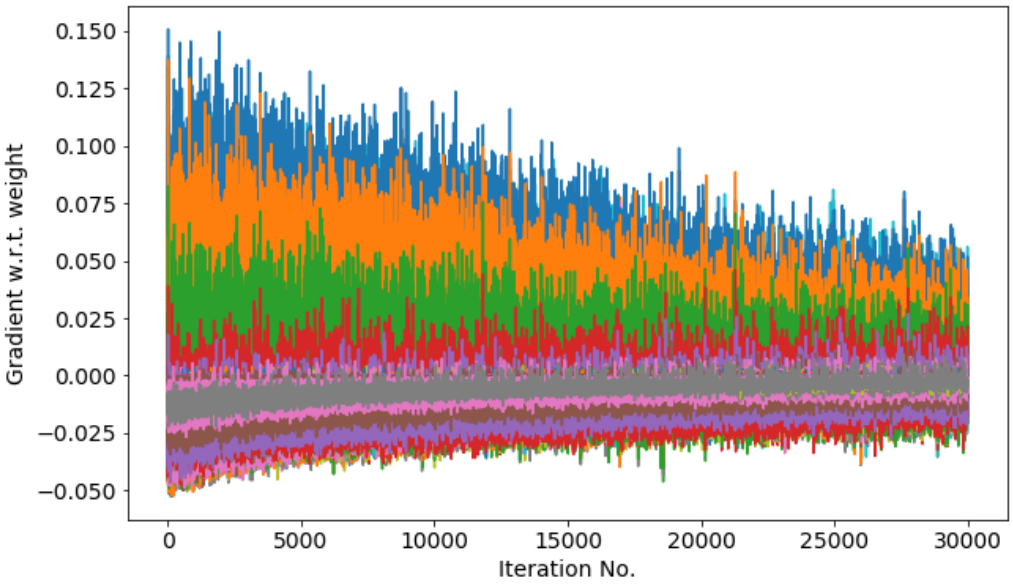


FIGURE 2.15: An illustration of the evolutionary track of the loss gradient with respect to each of the 10-neuron FNN weights. The figure shows 100 weight parameters randomly selected from the model.

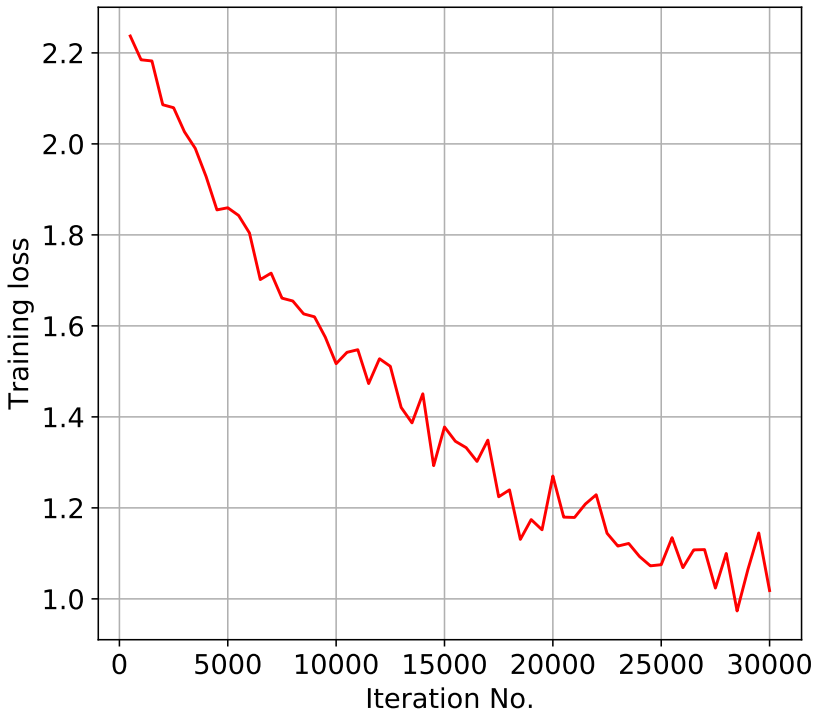


FIGURE 2.16: An example of the cross entropy loss curve when training the 10-neuron FNN.

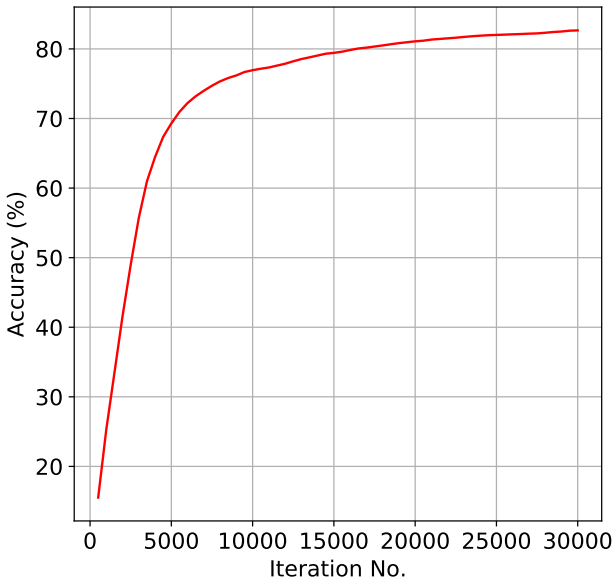


FIGURE 2.17: An illustration of the model learning curve, showing the evolution of model training accuracy as a percentage as the 10-neuron FNN trains with MNIST training data samples.

Besides tracking the evolution of model weight gradients and the model cross entropy loss, there is another metric frequently used in evaluating logistic regression model performance: **accuracy**, the percentage of correctly classified data samples.

Every 500 iterations of model training, I ask the model to make class predictions for 100 MNIST images chosen at random from the test set, and to compute the percentage of correct classifications. For each test image, the model will output a 10-element vector without normalization, extract its largest element, and re-assign it a value of 1. The script will then compare the one-hot image label vector and the predicted output vector, and decide if the model has made prediction consistent with its true label. The ratio of correct identifications to the total test sample size (100) multiplied by 100 will become the temporary (current) model test accuracy.

The evolution of model test accuracy is visualized in Figure 2.17 where it can be seen that the model has an initial prediction accuracy of less than 30%, but that this quickly climbs to above 70% within only 6 000 iterations. The accuracy growth gradually slows as the training continues, and reaches approximately 82% after 30 000 iterations of training.

2.4 Feedforward Neural Networks 2: Improving Model Performance

At this point, we have a simple and decent 10-neuron FNN for image digit class prediction. However, considering the best models in the literature, the performance of this one

is far from excellent. By changing the learning rate to a more aggressive value of 0.01 the model could reach 91% test accuracy, beyond this the model could also be improved by changing the parameter optimization method or the overall architecture. In this section, I will introduce several optimization methods and I will also discuss the possibility of adding additional 'hidden layers'.

2.4.1 Beyond SGD: other optimizers

As stated above, the learning ability of a model can be improved simply by changing the learning rate, but this must be done carefully: if the learning rate is too small, the model loss will converge slowly; when the learning rate is too large, the model loss can oscillate since the product of the gradient and the learning rate can be comparable to or even larger than the instantaneous model loss. How to deal with this issue becomes a challenge. One possibility is to replace the fixed learning rate with a learning rate schedule or, sometimes even better, a learning rate schedule for each model parameter. In this section I introduce three representative optimizers which implement this idea: the Adaptive Gradient algorithm (Adagrad; Duchi et al., 2011), the Root Mean Square Propagation method (RMSProp; Ruder, 2016), and the Adaptive Moment Estimation method (Adam; Kingma & Ba, 2014). These three methods all adapt their learning rates in accordance with the model parameter gradients.

- **Adagrad**: Adagrad is a modified SGD algorithm that aims to update each of the model parameters independently. Model parameters at iteration t would be updated in accordance with the parameter gradients of previous iterations. Extreme parameters would have their learning rates slow down, while those parameters with little update would receive higher learning rates.

  Compared to the standard mechanism of SGD, Adagrad adds a factor to the second term of Equation 2.20 such that

  $$P_{t+1} = P_t - \text{learningRate} \times g_t \times \left(\frac{1}{\sqrt{\sum_{\tau=1}^{t} g_t^2 + \epsilon}}\right), \tag{2.45}$$

  where t refers to the current iteration of training, and ϵ is a small number included to improve the numerical stability of the denominator. The quantity in the denominator is equivalent to the l2-norm of the parameter gradients (i.e. $\sqrt{\sum_{\tau=1}^{t} g_t^2 + \epsilon}$) (Duchi et al., 2011). By adding this factor, the optimization of each parameter at iteration t takes all previous gradients of the parameter into account, and balances the update steps across parameters.

  The introduction of the inverse l2-norm, however, has an apparent defect: model optimization will be aggressive in the early phases of training, while the rate of parameter optimization will quickly slow down as the l2-norm grows as a function

of iteration. In order to maintain efficient model training, the RMSProp algorithm was introduced.

- **RMSProp**: RMSProp is a modified SGD algorithm similar to Adagrad, first introduced by Ruder (2016). It aims to help those frequent model parameters to maintain a reasonably rapid update rate while also preventing the rapid growth of the parameter gradient l2-norm when using Adagrad.

  The idea is great, but how is this done? To achieve this, one needs to re-express the cumulative l2-norm of each parameter at each update, v_t, as:

$$v_t = v_{t-1} + g_t^2. \tag{2.46}$$

  Compared to Adagrad, RMSProp decays v_t by introducing a new constant β_1 such that

$$v_t = \beta_1 \times v_{t-1} + (1 - \beta_1) \times g_t^2. \tag{2.47}$$

- **Adam**: Adam (Kingma & Ba, 2014) is an update of RMSProp that replaces g_t with the first-order moment of the gradient, showing their cumulative history.

  The first-order moment of the gradient is expressed as:

$$m_t = \beta_2 \times m_{t-1} + (1 - \beta_2) \times g_t \tag{2.48}$$

  The inclusion of m_t monitors and controls the parameter optimization rate, as it balances the contribution from g_t with that of previous gradients. However, this is not the complete form of the optimizer. In practice, v_t and m_t will be divided by $(1 - \beta_1^t)$ and $(1 - \beta_2^t)$, respectively. Such a process is referred to as *bias correction*, converting the two terms into $\hat{v}_t$ and $\hat{m}_t$ and making the final parameter update rule of Adam:

$$P_{t+1} = P_t - \text{learningRate} \times \hat{m}_t \times (\frac{1}{\sqrt{\hat{v}_t} + \epsilon}). \tag{2.49}$$

  The introduction of the bias correction process starts from the initial bias of the two terms. At $t = 1$, v_1 and m_1 are equal to $(1 - \beta_2)\, g_1^2$ and $(1 - \beta_1)\, g_1$, where m_0 and v_0 are initialized to 0. However, the bias correction terms would 'correct' the two terms to become $\hat{v}_1 = g_1^2$ and $\hat{m}_1 = g_1$, respectively. In practice, β_2 is usually set to be closer to 1 than β_1. In `Pytorch` for example the default values are $\beta_1 = 0.9$, and $\beta_2 = 0.999$ (Paszke et al., 2019). If we substitute these two parameters into Equation 2.49 at $t = 1$, it becomes

$$P_1 = P_0 - \text{learningRate} \times g_1 \times (\frac{1}{\sqrt{g_1^2} + \epsilon}). \tag{2.50}$$

  Given that ϵ is a small value, we can easily see that by processing the bias correction, the specific customized value of β_1 and β_2 would have no impact on P_1. The bias

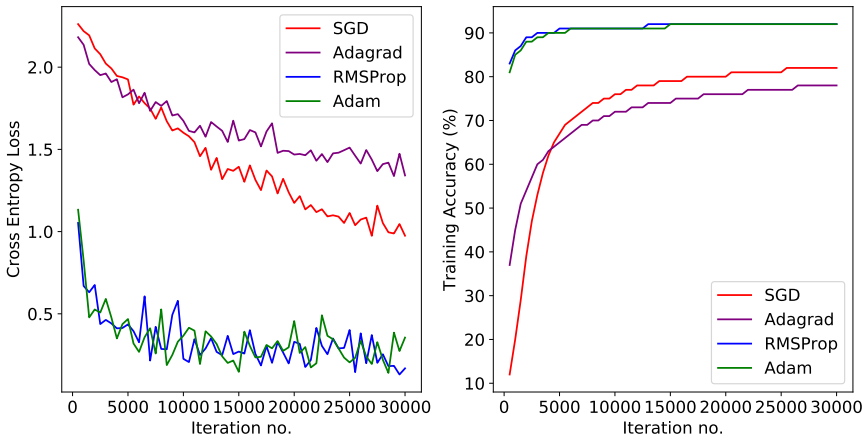


FIGURE 2.18: An illustration of the 10-neuron FNN model training cross entropy loss/ accuracy track (learning curves) when using SGD, Adagrad, RMSProp and Adam optimization method. The learning curves of models using the four methods are represented on the diagram in red, purple, blue and green, respectively.

corrected terms would then also have smaller influence upon the iterative terms, where P_2 can be seen as an example:

$$P_2 = P_1 - \text{learningRate} \times \frac{0.9\, g_1 + g_2}{0.9 + 1} \times \left(\frac{1}{\sqrt{\frac{0.999\, g_1^2 + g_2^2}{0.999+1}} + \epsilon}\right). \tag{2.51}$$

The model training losses and accuracies using each of the four optimisers (SGD, Adagrad, RMSProp and Adam), under the same model architecture as above and with an initial learning rate of 0.0001, are shown in Figure 2.18 as a function of iteration. The behaviour of the two metrics show that the SGD variants all have a stronger ability to converge faster at the beginning of model training. Models using either RMSProp or Adam show that the introduction of the 2nd order momentum accelerate the training convergence further and also extend the 'lifetime' of efficient training.

In the context of training accuracy, it can be seen that the models using RMSProp or Adam reach 91-92% within 10 000 iterations. However, one should also be aware that the same model using SGD can reach a similar accuracy if it uses a larger initial learning rate. This is already better than the result of 1-layer NN (linear classifier) proposed by Yann Lecun in 1998 (Lecun et al., 1998b).

2.4.2 Multi-layer Perceptrons

Apart from changing the loss optimizer, another approach to improve model performance is to build a Multi-Layer Perceptron (MLP; Hastie et al., 2001). MLPs are a class of FNN with at least one hidden layer between the input and output. MLPs are typically considered equivalent to FNNs with more than one hidden layer can be seen as MLP. Hidden layers, as the key building blocks of FNNs, can be described as:

- **Hidden Layer** In the context of neural networks, a hidden layer is found between the input and output of a model. Such a layer is similar to the output layer in structure, except that its outputs are passed through a non-linear **activation function** before entering the next layer.

In the context of biologically inspired neural networks, the activation function associated with a hidden layer can be described as an abstraction of the action potential rate for firing a cell (Hodgkin & Huxley, 1952). Practically, activation functions are able to map the outputs of a layer in a limited range (e.g. 0 to 1), and some activation functions can also normalize the outputs (e.g. Softmax, Sigmoid). In principle, activation functions can either be linear or non-linear, but usually non-linear activation functions are preferred as linear activation functions are unable to isolate parameters or maintain the complexity of a model in the same way as non-linear ones. Hidden layers with linear activation functions (e.g. Identity Knapp (2006)) can simply be seen as the linear inputs of the readout layer and the two layers can be rewritten as a single-layer neural network.

Coming back to non-linear activation functions, the family of most popular functions includes sigmoid (Mira & Hernández, 1995), tanh, and the widely adopted Rectified Linear Unit (ReLU; Hahnloser et al., 2000). which is described as:

$$f(x) = max(0, x), \tag{2.52}$$

for input x.
This simple form of activation function has a number of advantages such as:

- (1) Sparse activation: those inputs smaller than 0 will not be activated when doing model parameter back-propagation.
- (2) 'Gradient Vanishing' Prevention: compared to functions like the Sigmoid, the ReLU function prevents the 'gradient vanishing' problem that sometimes makes gradients too small to update the model weights.
- (3) Simple calculation: computation of the function only requires comparison, addition and multiplication.

Thanks to the advantages of this function, the efficient training of neural networks with more layers become possible. Although the ReLU also has some potential shortcomings, such as not being zero-centered and being unbounded, it has been shown in 2011 that the use of ReLU enables the training of deep supervised neural networks with no unsupervised pre-training (Glorot et al., 2011).

Hang on, **deep** supervised neural network? What does **deep** mean? Does it have any relationship to the term **Deep Learning** that we hear every day? In order to avoid confusion throughout the rest of this thesis, I here provide a general definition of some jargon:

- **Deep Learning**: a huge family of machine learning methods based on ANNs that use feature learning, irrespective of whether they are supervised, semi-supervised or unsupervised (Bengio et al., 2012; Schmidhuber, 2014; Goodfellow et al., 2016a).

```python
class LogisticRegressionModel(nn.Module):
    def __init__(self, input_dim, output_dim):
        super(LogisticRegressionModel, self).__init__()
        self.hidden_1 = nn.Linear(input_dim, 512)
        self.hidden_2 = nn.Linear(512,512)
        self.readout = nn.Linear(512,output_dim)
    def forward(self, x):
        x = F.relu(self.hidden_1(x))
        x = F.relu(self.hidden_2(x))
        out = self.readout(x)
        return out
```

FIGURE 2.19: The PYTHON `Pytorch` MLP model class foundation in the example.

- **Deep**: the adjective comes from the use of multiple layers in a neural network.
- **Deep Neural Network**: one of the sub-branches of Deep Learning. A DNN is an ANN with multiple layers between the input and output layers. MLPs with no fewer than 2 hidden layers are allowed to call themselves a DNN.

By understanding these concepts in advance, hopefully we can have more confidence when hearing or talking about them in daily life. In this thesis, all of the machine learning applications for radio astronomy use deep supervised neural networks and, considering its success in previous work (e.g. AlexNet; Krizhevsky et al. (2012b), ResNet; He et al. (2015), Inception-v4; Szegedy et al. (2016)), I also stick to using the ReLU as our activation function throughout the thesis.

Coming back to the MNIST classification problem, in the previous section we reached ∼92% model test accuracy using a 10-neuron FNN. Here I introduce another 1 or 2 hidden layers to the model architecture, adopt Adam as the training loss optimizer, and keep everything else unchanged.

In Figure 2.20 I show the `python` class setup for the 2-hidden layers FNN example. In comparison, when training the 1-hidden layer FNN model, I leave out the second hidden layer (`self.hidden_2`). The results of the three different models is visualised in Figure 2.20. It can be seen that as we add hidden layers and train the model for 30 000 iterations, the model cross entropy drops by two orders of magnitude, from more than 0.1 to less than 10^{-4}. Meanwhile, the improvement in test accuracy is also apparent, jumping from 92% to 98%. The 2-hidden layer model, although it has a similar final accuracy to that with 1 hidden layer, reaches 98% test accuracy at a faster rate. This improvement in model performance is largely due to the increased expressive ability given by the increased number of hidden layer parameters.

Besides a discussion of model performance between the three different architectures, the computational cost is also worth mentioning. When considering training time on the same machine, the 500-iteration computation time of the 10-neuron FNN is longer than those with hidden layers by around a factor of 3. The key to the smaller computation costs for these MLPs is the inclusion of the ReLU, because negative neuron outputs are neglected in model parameter gradient back-propagation. Their model training therefore becomes much sparser compared to that of the 10-neuron FNN.

At this point we have a decent FNN, already with 98% test accuracy. However, if we look at the model training mechanism for these 28 by 28 pixel images, we may find that

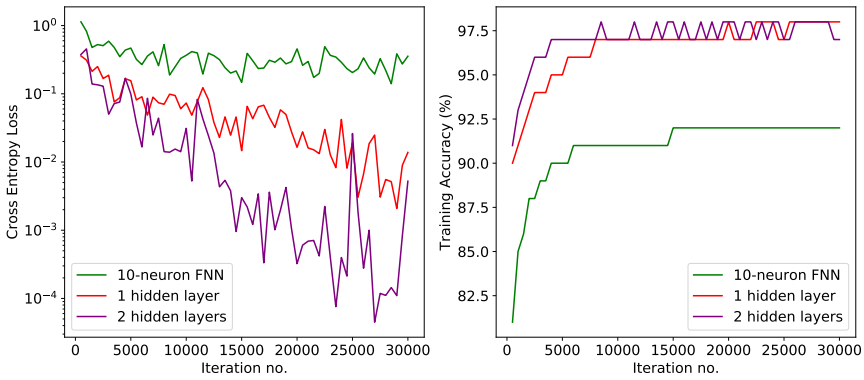


FIGURE 2.20: Model training cross entropy loss/testing accuracy comparisons for the three architectures used in this section. Left: The cross entropy loss evolution as a function of iteration. The loss is shown on a log scale. Right: The model training accuracy as a function of iteration.

it is different from our own daily habits. The way that humans recognize hand-written digits from an image is to see them as images, rather than seeing them as squeezed one-dimensional arrays, and images, depending on the color scale (e.g. grey, RGB), have at least 2 or more dimensions. Do we have any neural network architectures that are similar to our visual system and might classify MNIST digits better? Convolutional Neural Networks (CNN; Lecun et al., 1998b) could be an good option.

2.5 Convolutional Neural Networks 1: Origin - Neocognitron

Before talking about what Convolutional Neural Networks (CNNs) actually are, let's first look at an inspirational study of the striate cortex of Macaque and Spider monkeys (Hubel & Wiesel, 1968). This study found that when the retina of a monkey is stimulated by a light spot or pattern, the cells behind each of its receptive fields would be activated and transmit information. The simple receptive field of these monkeys ranged from 0.25 $\times$ 0.25° to 0.25 $\times$ 0.75°, while most field-of-view extents were much larger at 5 - 10° (Hubel & Wiesel, 1968). Also, depending on different cells behind the fields, some fields might be placed side by side, or have gaps in between (Hubel & Wiesel, 1968). In other words, it took hundreds of receptive fields all together to assemble the FOV of spider monkeys. Another discovery found in this study is that the receptive fields of monkeys had higher sensitivity to changes in light stimulus orientation compared to cats, and that a small proportion of their cells were colour coded (Hubel & Wiesel, 1968). The receptive fields should therefore be able to keep both positional and colour information.

If one wanted to build an artificial neural network that had the potential to imitate a monkey or a cat's visual cortex, some of its layers at least should be able to replicate the visual cortex functionalities identified above. In the example of the 10-neuron FNN, each MNIST image was reshaped to a vector and imported to each neuron independently.

Such a layer setup would fail to reproduce the receptive fields as it loses positional information for each image pixel, and might lose image colour information if the input image has colour. In principle, one could overcome these issues by:

1. Receptive field reconstruction: rather than importing all image inputs to each neuron, only import image inputs from customized receptive fields to a specified neuron. Receptive fields could be overlapped, side-by-side, or even have gaps in between.

2. Retina - receptive field correspondence: in order to keep the information on relative positions between receptive fields the arrangement of neurons in a layer should be consistent with the positional import sequence of the input image.

3. Colour information maintenance: rather than reshaping the image input, each neuron in the layer should be able to import all input values within the specified receptive field, regardless of the number of colour channels (e,g. 1 channel for greyscale images, 3 for RGB images). In other words, a neuron in the layer should be able to import input volumes.

When we look into the first three issues, they can all be solved by resetting the neuron arrangement and redefining the input sequence rules. However, this will inevitably change the way that we sum the product of the model weights and the input values, as the layer output would be at least 2 dimensional. In addition to this, the relationship between the input and output of each neuron would look like:

$$y = \left(\sum_{i=1}^{\text{width}} \sum_{j=1}^{\text{height}} \sum_{k=1}^{\text{channel}} w_{i,j,k}\, x_{i,j,k} \right) + b, \tag{2.53}$$

where **width** and **height** here refer to the width and height of the specified receptive field, and **channel** represents the number of input channels in the input image (e.g. 1 for greyscale images, 3 for RGB images). Such an operation requires the neuron to consider inputs in three dimensions rather than two.

If we look back at historical studies, we can find that this is similar to the 'S-cell' setup of Fukushima's **neocognitron**: a hierarchical neural network capable of visual pattern recognition directly inspired by Hubel's study (Fukushima, 1980). Indeed, neocognitron is often considered to be the foundation of CNNs.

The 'Simple Cell' (S-cell) proposed by Fukushima adopted a neuron form similar to that of Equation 2.53. S-cells are arranged in a plane with customized widths and heights and, furthermore, it is allowed to have more than one S-cell plane stacked together in each hidden layer. Same inputs are then allowed to have multiple corresponding weights, which increases the expressive ability of the hidden layer significantly. The number of neuron layers within a hidden layer is referred to as the **depth**. When Fukushima's team was building a network for hand-written number recognition, for instance, they initially imported sampled greyscale images of 19×19 pixels into S-cell planes of size 19×19, with 12 planes stacked together (Fukushima, 1980).

Hidden layers of this type, what I have called S-cell layers here, form the rudimentary building blocks of **convolutional layers**, the core of a CNN. Each S-cell plane is able to extract features from inputs generated in the previous layer, and helps to recognize image class, e.g. 0-9 for hand-written digits. According to the findings of Fukushima, S-cell layers at higher stages extract global features and in the lower stages the S-cell layers are able to extract local features (Fukushima, 1980). Since such hidden layers still differ from modern convolutional layers, I will outline their differences in the next section. Interestingly, the outputs of S-cell neurons are also passed through a ReLU function, which is the same as the hidden layers in the previous FNN examples.

In addition to S-cells being able to extract local features from input images, the other contribution Fukushima made in this study was the invention of the 'Complex-cell' (C-cell). The functionality of C-cells is to process the information of S-cell outputs and perform down-sampling. The invention of C-cells largely inspired the later development of down-sampling layers such as **pooling layers** and information layers such as **fully-connected layers**. We will explain and summarize these layers in the next section, where we will formally introduce the building blocks of modern CNNs.

2.6 Convolutional Neural Networks 2: Building Blocks of a CNN

Generally speaking, a conventional CNN usually includes the following building blocks:

1. convolutional layers
2. fully-connected layers
3. pooling layers
4. activation layers
5. readout layers (fully-connected layers without activation)

Since I have introduced activation functions already, I will introduce $(1) - (3)$ in this section in sequence and show their correlation with historic research. I will cover (5) when introducing fully-connected layers.

2.6.1 (1) Convolutional Layer

The neocognitron had by no doubt taken a big step forward, from traditional FNNs to a network closer to that of a Spider Monkey's visual cortex. However, S-cell planes still have a few shortages, which could be improved. For example:

- Positional constraints on feature detection: although S-cells are able to extract features from inputs, it is difficult for them to recognize a particular feature at any location in an input field.
- Too many parameters: for every single S-cell all the inputs need to have independent weight parameters, which requires gigantic numbers of model parameters.

Could we find a mathematical operation to help S-cell planes overcome these shortages? In response to this question, Yann Lecun and his colleagues introduced *weight-sharing* (Rumelhart et al., 1986; LeCun et al., 1989), allowing the network to recognize the same features at different locations in an image with only a small number of neurons in a plane (LeCun et al., 1989). In practice, this is achieved through **convolution**, which results in the foundation of the convolutional layer, the core of a convolutional network as they named it (LeCun et al., 1989; Lecun et al., 1998b):

Convolution is a mathematical operation between two functions (say f and g) that produces a third function showing how the shape of one function is modified by the other. It is classically defined as the integral of the product of f and g after one of them is shifted and reversed. A convolutional layer's parameters consist of multiple learnable filters. Each filter can be seen as a weight parameter plane, extended across the full depth of the input volume, convolved across the width and height of of input volume, which calculates the dot product of the filter weights and the input. After activation, the output forms a feature map (or activation map). The network then extracts its primary features from arbitrary positions of input images.

Lecun and his colleagues introduced the convolution operation to the S-cell model. Rather than only using the concept of a receptive field when importing inputs to neurons, each filter in a convolutional layer no longer sees the complete visual field, but instead a virtual receptive field with given width, height and channel depth is weight-sharing when performing convolution. An example is shown in Figure 2.21 where the input has a volume of $7 \times 7 \times 1$ (greyscale image) and the convolutional layer has 2 filters with identical sizes of width = 3, height = 3 and channels = 1. Starting from the first 3 by 3 pixel area, each channel of a filter is convolved with the corresponding channel of the 3 by 3 pixel field, has the dot product calculated, and then adds the channel's bias. For multi-channel inputs, these outputs are then summed with the other channel outputs of the filter and become the first element in the output volume. Similar operations can be done using a stride of 2 in the x direction, with one column of input overlapped. When the convolution of the first three rows is finished, the second filter would start from row 1, moves down with a stride of 2, and repeat the operations. After all the operations are complete, the convolutional layer produces an output volume of $3 \times 3 \times 2$. Such operations therefore decrease the required number of neurons in the following layer.

Knowing how the filters in the convolutional layer operate, we can now try to visualize how a feature is extracted from the input by a filter. For simplicity we here ignore the bias parameter. Assuming the extracted feature is a line, see Figure 2.22, and that there are two input receptive fields extracted at arbitrary positions within a larger visual field, showing character 'I' and 'L'. By performing the same operation as described above, receptive field 1 would result in an output of 4 and receptive field 2 would return an output of 1. The feature map produced by the filter is more appropriate for recognizing the character 'I', whereas the identification of 'L', on the other hand, may require the involvement of other filters.

FIGURE 2.21: Left: Schematic diagram of CNN neurons. Green cuboid represents a segment of a convolutional layer, while blue spheres within the cuboid are CNN neurons. w, h and d of the cuboid is width, height, and depth of a layer. Right: Convolution process of a ($3 \times 3 \times 2$) convolutional layer. Two 3×3 matrices within black boxes convolve with each other and plus constant bias b0 equals 1, the value locates at the [0,0,1] of output volume.

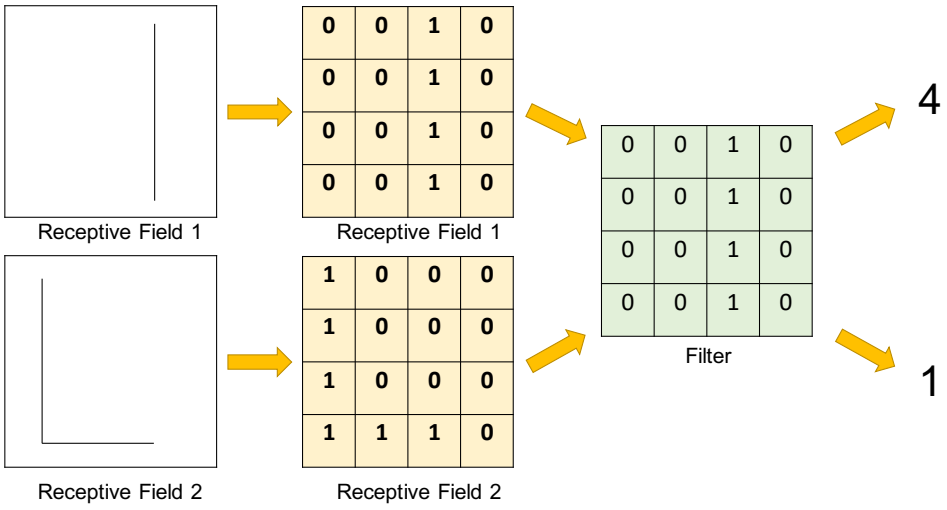


FIGURE 2.22: An illustration of how a specific filter in a convolutional layer decides which receptive field on the diagram (representing letter 'I' and 'L') is more similar to what it 'learned'. From left to right: (i) Two receptive fields with pixel size of 4×4; (ii) The same fields represented in the matrix form. The pattern shown in each receptive field would be given a value of 1 in their primary location, where the empty spaces would be numbered 0. The numerical receptive fields could then be 'convolved' with (iii) the filter function that also in the matrix form and (d) output the resulting number. By comparing the resulting numbers, one could judge which receptive field fits the given filter function better.

2.6.2 (2) Fully-connected layer

Although the name is different, a fully-connected layer is equivalent to the hidden layer of an FNN. The neurons of a fully-connected layer are fully-connected to the neurons in the previous layer. LeCun et al. (1989), as a pioneer example, have its third and fourth hidden layers fully connected to its previous layer neurons. We here re-introduce the term, as the top, middle, and bottom fully connected layers have similar but different-functionalities. The differences are:

- Top: the earliest fully-connected layer serves as the bridge between the outputs of convolutional/pooling layers and the subsequent fully-connected layers. It flattens the features extracted by the convolutional/pooling layers. In extreme cases, this layer can also serve as the readout layer.
- Middle: the same as hidden layers in FNNs, abstract or split features are imported from the previous layer. It is not essential to include these layers.
- Bottom: this layer is also called the loss or readout layer. Usually it is the final layer of the full network, computing the model loss and providing a numerical probabilistic prediction.

The neurons in fully-connected layers are learnable, and are able to convert extracted features into a final prediction. The functionalities of fully connected layers, however, are not limited to this. We will discuss them further as we meet specific issues in later sections.

2.6.3 (3) Pooling layer

Although convolutional layers are able to extract features from their inputs, the output feature maps are sensitive to the feature locations in the input. That is, if the aforementioned 'I' is at the top left corner of the image, and was identified by the filter, the activated neurons would be at the top left corner of the output feature map. If the input 'I' is at the image center, the corresponding position on the feature map would be at its center. This positional correspondence is maintained throughout the lower layers as well. In order to decrease the influence of local feature translation within the feature maps of the lower layers, a representative approach is pooling:

Pooling A layer performs non-linear down-sampling operations. It partitions an input into non-overlapping rectangular segments and produces summarized scalar outputs for each segment.

By summarizing patches of features, the pooling layers reduce the size of the feature space, the number of model parameters, and the computational cost of the network. Such an operation can be seen as similar to convolution, with the exception that the pooling layers are not learnable. There are two common types of pooling operation:

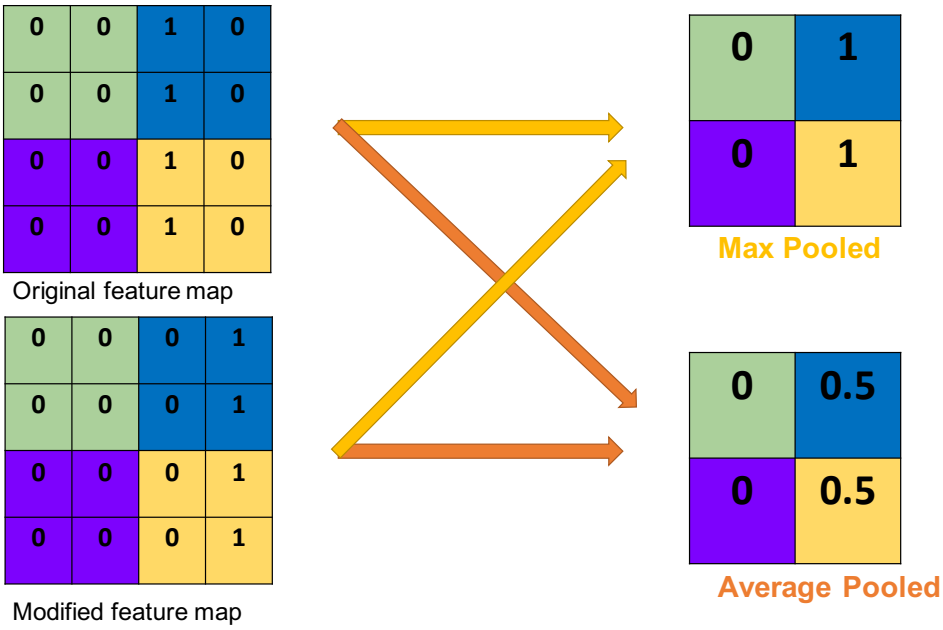


FIGURE 2.23: An illustration of how Max-pooling and Average-pooling layer works. `Upper left`: The original feature map with a size of 4×4, where the ones are located at the third column; `Lower left`: The modified feature map of the same size, while the ones are translated to the fourth column. `Upper right`: the max-pooled outcome operated on both feature maps, along with both kernel size and stride as 2. For a patch of each of the feature map with identical color, the max-pooling function would ouput the maximum value in the region, which resulting a down-sampled 2×2 output. `Lower right`: A similar down-sampled output as of the upper right one, while the outputs are the averaged values of the same input regions.

Average Pooling Average Pooling returns the average value of each patch of the feature map.

Max Pooling Max Pooling returns the maximum value in each patch of the feature map.

Figure 2.23 is an illustration of these two pooling operations, using a 4×4 feature map with the same values as receptive field 1 mentioned earlier. The two example operations both summarize outputs from non-overlapped 2×2 patches of the feature maps.

Although average pooling was used as a sub-sampling tool in early CNN applications (Lecun et al., 1998b), it was later found that in the regime of computer vision tasks such as image classification that max pooling would perform better:

"In a nutshell, the reason is that features tend to encode the spatial presence of some pattern or concept over the different tiles of the feature map (hence, the term feature map), and it's more informative to look at the maximum features than at their average presence." Chollet (2017)

Considering the advantages of max pooling, I will stick to the use of max pooling for the rest of the work presented here. In the next section, we will explore a pioneering and simple CNN: LeNet-5 (LeCun et al., 1989; Lecun et al., 1998b).

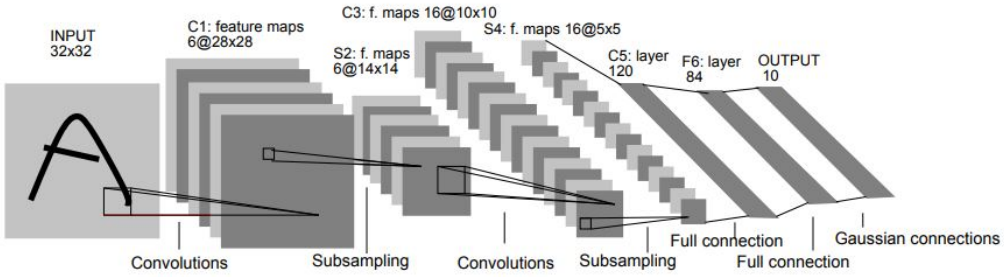


FIGURE 2.24: The Figure 2 of Lecun et al. (1998b), which shows the architecture of LeNet-5. Each plane on the diagram represents a feature map.

Layer No.	Layer Type	Input Channel	Output Channel	Kernel Size	Stride	Activation
1	Convolutional	1	6	5	1	Tanh
2	Average Pooling	6	6	2	2	
3	Convolutional	6	16	5	1	Tanh
4	Average Pooling	16	16	2	2	
5	Convolutional	16	120	5	1	Tanh
6	Fully-connected	120	84			Tanh
7	Fully-connected	84	10			RBF

TABLE 2.1: A summary of LeNet-5 architecture.

2.7 Convolutional Neural Networks 3: LeNet-5

LeNet-5 is a 7-layer CNN (not including the input layer), and was developed from LeNet-1. LeNet-1 was built with only two convolutional layers, two fully-connected layers, and a final readout layer (LeCun et al., 1989). Continuous work in the field (e.g. LeCun et al., 1989; Le Cun et al., 1989) and comparison with various methods of handwritten digit recognition finally led to the foundation of LeNet-5 (Lecun et al., 1998b), which was found to outperform other models in the same period (Lecun et al., 1998b).

The architecture of LeNet-5 is shown in Figure 2.24 and summarised in Table 2.1. Compared to LeNet-1, LeNet-5 introduced two additional average pooling layers and used the Tanh function as the activation for both its convolutional layers and its top fully-connected layer. Moreover, the final layer of LeNet-5 is composed of Euclidean Radial Basis Function (RBF) units, where each RBF unit calculates the Euclidean distance between its input label and the corresponding output vector. In other words, if an input is far from its output, the RBF value would be large (Lecun et al., 1998b).

The inputs introduced to the network were 32×32 pixels, larger than the original MNIST images as the distinctive features then could appear in the center of the receptive field of the top feature detectors (Lecun et al., 1998b). Finally, LeNet-5 used the Maximum Likelihood Estimation Criterion (MLE) as a loss function, which in their case was equivalent to the Mean Squared Error (MSE).

In this section, I run a modified LeNet-5 for classifying MNIST handwritten digits, which is also the training/testing database LeNet used. These modifications are made to convert LeNet-5 into a network closer to recent CNNs. The modifications are:

```
class Modified_Lenet_5(nn.Module):
    def __init__(self):
        super(Modified_Lenet_5, self).__init__()
        # ReLU
        self.relu = nn.ReLU()
        # Max pool
        self.maxpool = nn.MaxPool2d(kernel_size=2)
        # Convolution 1
        self.cnn1 = nn.Conv2d(in_channels=1, out_channels=6, kernel_size=5, stride=1,padding=2)
        # Convolution 2
        self.cnn2 = nn.Conv2d(in_channels=6, out_channels=16, kernel_size=5, stride=1)
        # Convolution 3
        self.cnn3 = nn.Conv2d(in_channels=16, out_channels=120, kernel_size=5, stride=1)
        # Fully connected 1
        self.fc1 = nn.Linear(120, 84)
        # Fully connected 2 (Readout)
        self.fc2 = nn.Linear(84,10)

    def forward(self, x):
        # Convolution 1
        out = self.cnn1(x)
        out = self.relu(out)
        # Max pool 1
        out = self.maxpool(out)
        # Convolution 2
        out = self.cnn2(out)
        out = self.relu(out)
        # Max pool 2
        out = self.maxpool(out)
        # Convolution 3
        out = self.cnn3(out)
        out = self.relu(out)
        # Resize
        out = out.view(out.size(0), -1)
        # fully conntected 1
        out = self.fc1(out)
        out = self.relu(out)
        # fully connected 2 (Readout)
        out = self.fc2(out)
        return out
```

FIGURE 2.25: A Pytorch model class setup of the modified LeNet-5 in our example.

1. I replace the average pooling layers with max pooling layers, leaving the kernel size and the stride size unchanged.

2. I replace the Tanh activation function with the ReLU function.

3. I replace the RBF method in the final layer with the Softmax function.

4. Rather than using MLE loss function, I adopt the Cross Entropy loss function in order to compute digit class probability distribution.

These modifications result in a `Pytorch` model class as shown in Figure 2.25. In the original LeNet-5, the image inputs had a size of 32×32 pixels, which requires us to resize the MNIST images before importing them into the model. If one did not wish to resize the inputs, one could instead apply zero-padding.

Zero-Padding In the context of machine learning, zero-padding refers to adding zeros around a given matrix, helping to preserve features surrounding the the edges of the primary matrix.

Compared to resizing the image input, padding can retain the completeness of input information and be used in arbitrary convolutional layers. In our example, by defining the width of the padding as 2, each 28×28 pixel input would have 2 rows of zero-filled pixels on each of its sides, making the final input size 32×32 pixels. If we have an input size I, a kernel size K, a stride S and padding P in a specific layer then the output size O can be described as:

$$O = \frac{I + 2P - (K-1) - 1}{S} + 1 \tag{2.54}$$

```
from torchsummary import summary
summary(model,(1,28,28))
```

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1            [-1, 6, 28, 28]             156
              ReLU-2            [-1, 6, 28, 28]               0
         MaxPool2d-3            [-1, 6, 14, 14]               0
            Conv2d-4           [-1, 16, 10, 10]           2,416
              ReLU-5           [-1, 16, 10, 10]               0
         MaxPool2d-6             [-1, 16, 5, 5]               0
            Conv2d-7            [-1, 120, 1, 1]          48,120
              ReLU-8            [-1, 120, 1, 1]               0
            Linear-9                  [-1, 84]          10,164
             ReLU-10                  [-1, 84]               0
           Linear-11                  [-1, 10]             850
================================================================
Total params: 61,706
Trainable params: 61,706
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.11
Params size (MB): 0.24
Estimated Total Size (MB): 0.35
----------------------------------------------------------------
```

FIGURE 2.26: The model parameter summary of the modified LeNet-5 model in our example. This is achieved by using PYTHON *torchsummary.summary()* function, with given data input of size (1,28,28).

For instance, the first convolutional layer in our example has $I = 28$, $K = 5$, $S = 1$ and $P = 2$, which gives $O = 28$. By using the above equation, one could receive the same result as that given by the `torch.summary.summary` function, see Figure 2.26, a `Pytorch` model summary function. In practice, one can use this function to facilitate examining the correspondence of input/output layer shapes.

As well as summarizing the output shape of each model layer, this function also calculates the number of trainable parameters per layer. The top convolutional layer, for example, has a kernel size 5 and 6 filters, resulting in $5 \times 5 \times 6 = 150$ weight parameters and 6 bias parameters, giving a total number of 156 parameters.

Now knowing the details of the modified LeNet-5 model, we can now start to train the model using the hyper-parameters inherited from the previous example. The model training results can be seen in Figure 2.27, where I show the model training loss, the test accuracy, and also the test loss. It can be seen that both training and test loss generally decrease monotonically, while hints of **over-fitting** are becoming apparent when the training is about to finish.

Over-fitting Over-fitting refers to the situation where the model does not learn general features, but instead memorizes almost everything in the training data.

When a model is over-fitting, its test loss might saturate or even increase. Meanwhile, the training loss of the model will keep dropping. For instance, if over-fitting takes place after around 25 000 iterations, one would observe the testing loss curve of the modified LeNet-5 bounce back: testing loss start to increase at 25 000 th iteration. In practice, over-fitting can be prevented or partly reduced by using regularization methods such as **validation based early stopping**, **dropout** and **batch normalization**.

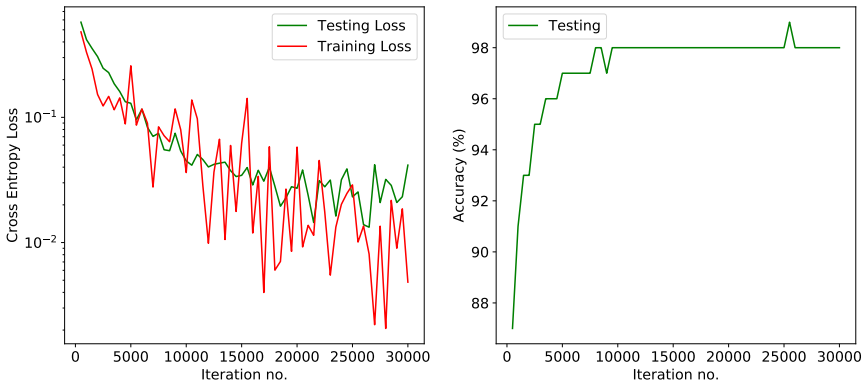


FIGURE 2.27: An illustration of the modified LeNet-5 model learning curves. Left: The model training/testing cross entropy loss curve, where training loss is colored in red and testing loss is in green color. Right: The model testing accuracy curve in green color.

Validation Based Early Stopping This regularization method splits the primary training set into a new smaller training set and a validation set. The validation loss is seen as a proxy for the generalization error in estimating the start of over-fitting (Prechelt, 1996).

Dropout A method to randomly drop network units and their connections when the model is training. The dropout samples form an exponential number of different "thinned" networks. When doing testing, the model will adopt a single unthinned network of smaller weights to approximate the averaging effect (Srivastava et al., 2014).

Batch Normalization Initially proposed to alleviate internal covariate shifts, this regularization method normalizes layer outputs for each data batch. This method is found to be able to improve the generalization properties of a network, speed up the training process, and partly regularize a network.

In our example, we split the 60 000 MNIST training samples randomly into a new training set of 50 000 objects and a validation set of 10 000 objects. Such an approach ensures that the training and validation sets have high-level similarity, and one can recognize model over-fitting when the validation loss of the network no longer follows or moves away from the decreasing training loss. When such a phenomena happens, an early-stopping strategy might be adopted to have the model stop training before the validation loss starts to increase.

Dropout, on the other hand, randomly zeros outputs from a specified layer with a customized probability, p, and scales the output by a factor of $1/1-p$ (Paszke et al., 2019). For the rest of this section, I adopt $p = 0.2$ whenever I apply the dropout function.

Finally, batch normalization is used to minimize the effect of random parameter initialization on the batch input distribution. Such an effect is also called internal covariate shift. Considering a batch size of 100, we here list the procedure for batch normalization

at each neuron output (Ioffe & Szegedy, 2015):

$$\mu_B = \sum_{i=1}^{100} \frac{1}{100} x_i; \quad \sigma_B^2 = \frac{1}{100} \sum_{i=1}^{100} (x_i - \mu_B)^2 \tag{2.55}$$

$$y_i = \gamma \hat{x}_i + \beta = \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \tag{2.56}$$

where y_i refers to the batch normalized neuron output. If a layer has k feature maps, batch normalization would be independently processed on each feature map, and the Equation 2.56 and for the k^{th} feature map becomes:

$$y_i^{(k)} = \gamma^{(k)} \hat{x}_i^{(k)} + \beta^{(k)} = \gamma^{(k)} \frac{x_i^{(k)} - \mu_B^{(k)}}{\sqrt{(\sigma_B^{(k)})^2 + \epsilon}} + \beta^{(k)}. \tag{2.57}$$

Different from the first two regularization methods, batch normalization functions provide learnable parameters: γ and β. Each feature map has one γ and β, usually initialized as 1 and 0. These parameters could be optimized via back-propagation, in principle using the following derivatives:

$$\frac{\partial L}{\partial \beta} = \sum_i^{100} \frac{\partial L}{\partial y_i} \frac{\partial y_i}{\partial \beta} = \sum_i^{100} \frac{\partial L}{\partial y_i} \tag{2.58}$$

$$\frac{\partial L}{\partial \gamma} = \sum_i^{100} \frac{\partial L}{\partial y_i} \frac{\partial y_i}{\partial \gamma} = \sum_i^{100} \frac{\partial L}{\partial y_i} \hat{x}_i \tag{2.59}$$

The updates to input x_i, on the other hand, require the following derivatives:

$$\frac{\partial L}{\partial \hat{x}_i} = \sum_i^{100} \frac{\partial L}{\partial y_i} \frac{\partial y_i}{\partial \hat{x}_i} = \sum_i^{100} \frac{\partial L}{\partial y_i} \gamma, \tag{2.60}$$

$$\frac{\partial L}{\partial \mu_B} = \sum_i^{100} \frac{\partial L}{\partial \hat{x}_i} \frac{\partial \hat{x}_i}{\partial \mu_B} + \frac{\partial L}{\partial {\sigma_B}^2} \frac{\partial {\sigma_B}^2}{\partial \mu_B}, \tag{2.61}$$

$$\frac{\partial L}{\partial {\sigma_B}^2} = \sum_i^{100} \frac{\partial L}{\partial \hat{x}_i} \frac{\partial \hat{x}_i}{\partial {\sigma_B}^2} = \sum_i^{100} \frac{\partial L}{\partial \hat{x}_i} \frac{-(x_i - \mu_B)}{2({\sigma_B}^2 + \epsilon)^{3/2}}. \tag{2.62}$$

Equations. 2.60-2.62 then pave the way for solving $\frac{\partial L}{\partial x_i}$:

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial \hat{x}_i} \frac{1}{\sqrt{{\sigma_B}^2 + \epsilon}} + \frac{\partial L}{\partial {\sigma_B}^2} \frac{2(x_i - \mu_B)}{100} + \frac{\partial L}{\partial \mu_B} \frac{1}{100}. \tag{2.63}$$

Since dropout and batch normalization were invented, dropout has been implemented in image classification (Szegedy et al., 2014), speech recognition (Hannun et al., 2014), and natural language processing (Kim et al., 2015). On the other hand, batch normalization has also been applied in many recent approaches (Szegedy et al., 2015, 2016; Howard et al., 2017). Interestingly, although the effectiveness of the two functions has

been proved, exactly where and how to use them is still under discussion. To date, there are several strategies performed/proposed:

1. Batch norm before ReLU: This method was proposed at the invention of batch norm (Ioffe & Szegedy, 2015). Recent successful architectures, including ResNet (He et al., 2015), have been implemented using batch norm without dropout involvement.

2. 'half' batch norm + 'half' dropout after ReLU: A modified AlexNet architecture (Krizhevsky et al., 2012b) applied batch norm to the activated convolutional layer outputs, and dropout on the activated fully-connected layer outputs, achieving human-comparable radio galaxy morphology classification accuracy (Aniyan & Thorat, 2017).

3. weight $\rightarrow$ batch norm $\rightarrow$ dropout $\rightarrow$ ReLU: Li et al. (2018) discussed the possibility of using batch normalization and dropout after a weight layer sequentially and suggested this strategy. The alternative dropout $\rightarrow$ batch normalization strategy was not preferred as dropout would scale neuron responses by its retaining ratio p, and thus change the neuron variance as the training processes. However, the batch normalization would still retain the statistical moving variance of the original neuron outputs (Li et al., 2018).

4. batch norm $\rightarrow$ dropout $\rightarrow$ weight $\rightarrow$ ReLU: Another analysis discussed the disadvantage of placing batch norm before the ReLU activation function, as the non-negative ReLU responses would make the layers weight train in a sub-optimal way (Chen et al., 2019). They then named this combination of the batch norm layer and the dropout layer as an Independent Component (IC) layer, and further proposed this strategy to achieve better model performance.

In order to compare these approaches, we apply each strategy on the modified LeNet-5 model, with an initial learning rate of 0.01 and using SGD optimization, see Figure 2.28. Dropout layers are used with a retaining percentage of 20%. We then examine them in the terms of model losses and convergence speed.

In the context of model loss, it can be seen that these approaches have all achieved reasonable learning outcomes. The validation losses have generally followed similarly decreasing paths to their corresponding training losses, as have the test losses. Compared to the primary model, those approaches using strategies (2)-(4) have provided lower model test losses. When considering over-fitting, it is interesting to see that having batch norm after ReLU increases the over-fitting of a model, a situation which could be alleviated by introducing dropout layers either after the batch norm layers or the activated fully-connected layers.

Looking at the convergence speed, one can see that the involvement of batch normalization does speed up the process, while the inclusion of dropout slows the process. Interestingly, by using IC layers, a model can largely maintain its unregularized convergence speed. In practice, developers would need to evaluate which approach they prefer,

as the model performance might vary when training with different data sets or network architectures.

2.8 Convolutional Neural Networks 4: Beyond classic architecture

From the discussion in Sections 2.5 - 2.7, we know that the convolutional layers of a CNN are able to extract features from image inputs with limited or even no image preprocessing, and produce extracted features that can be used to make model predictions. We also learned that the upper convolutional layers in a model tend to learn global features, while the lower ones learn more specific image features (Fukushima, 1980). The characteristics of such convolutional layers can also lead to further applications in the context of training strategy and network architecture. In this section, I introduce how models with convolutional layers can be used to apply a **transfer learning** approach (e.g. Pratt, 1993; Pan & Yang, 2010), and how classical CNNs can be modified to become **Multi-branch CNNs** (e.g. Li et al., 2017; Aslani et al., 2018; Georgakilas et al., 2020), both of which will be implemented later in this thesis.

2.8.1 Transfer learning

In general, *transfer learning* is more of a research problem on how to 'inherit' knowledge from a solved problem and apply it to novel applications (Pratt, 1993). Such a new problem might be relevant or irrelevant to the already solved problem. In the context of generalization, it is said that transfer learning is able to use what has been learned in one setting to improve generalization in another setting (Goodfellow et al., 2016a). For instance, one could have a model trained to learn general features from handwritten digits and apply it to handwritten character recognition (Maitra et al., 2015).

In order to understand different possible transfer learning strategies, we follow Pan & Yang (2010) and define **domain** and **task**:

- Domain: A domain **D** includes a feature space X and a marginal probability distribution **P(X)**, where X = $x_1, ..., x_n \in X$. A domain can be denoted as **D** = $\{X$, **P(X)**$\}$.
- Task: Given a specific domain **D**, a task T includes a label space Y and a objective predictive function f(·) that can learn from the training data. The function f(·) consists of pairs $\{x_i, y_i\}$, where $x_i \in$ X and $y_i \in Y$. A task can be denoted as T = $\{Y$, **f(·)**$\}$.

What do these parameters actually mean? Considering the logistic regression example of fever detection using mercurial thermometer, we list the brief explanation of these parameters as follows (Pan & Yang, 2010):

- **X**: a particular learning sample. The 70 temperature data points in Section 2.2 together build the learning sample in the example.

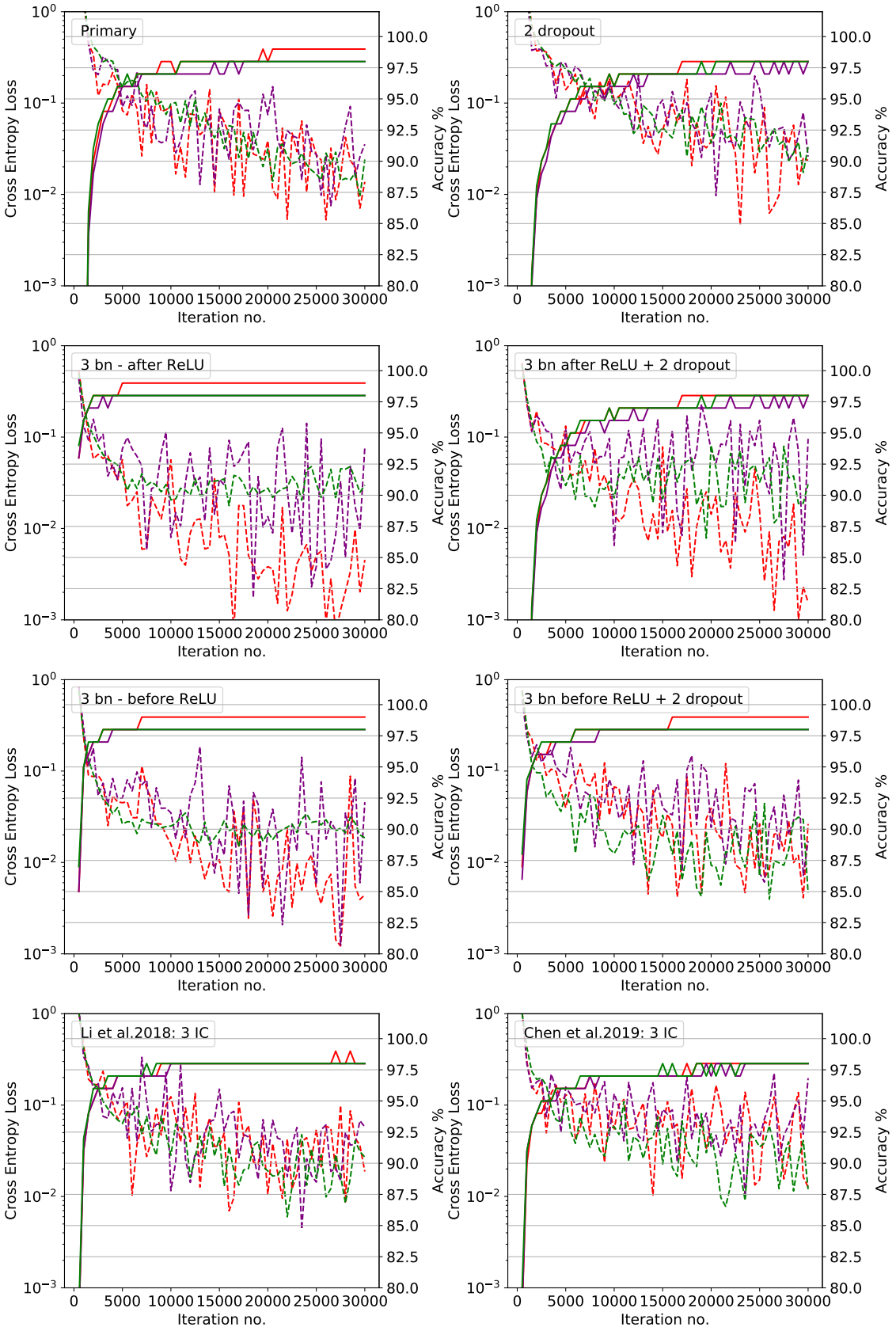


FIGURE 2.28: The Learning curves of the modified LeNet-5 architecture trained with different model regularization strategies. The solid and dashed lines on the diagram refers to testing accuracy and cross entropy loss, respectively. Red/purple/green color on the diagram identically represents to the training/validation/testing learning curve.

- x_i: The i_{th} feature vector of **X**. In machine learning, a feature vector comprises a number of numerical features that represent an object. In the case of fever detection, the author's body temperature (e.g. 36.4°) can be seen as the first and only feature of the feature vector $\{x_1\}$, where x_1 = [36.4].
- X: The vector space associated with all of these feature vectors is named as a feature space X. In this example, $x_1, ..., x_{70} \in X$.
- **P(X)**: The marginal probability distribution of the training sample **X**, i.e. the probability distribution of the 70 temperature data points in this sample, when other features (although nonexistent in this example) are not taken into account.
- y_i: Labels. In this problem, True or False, referring to whether one's body temperature hits the fever limit.
- Y: The set of all labels.
- f(·): can be used to predict the corresponding probability of a new instance x. The function can also be written as **P(y|x)**. In this example, it refers to the predicted probability of having a fever with a given temperature value.

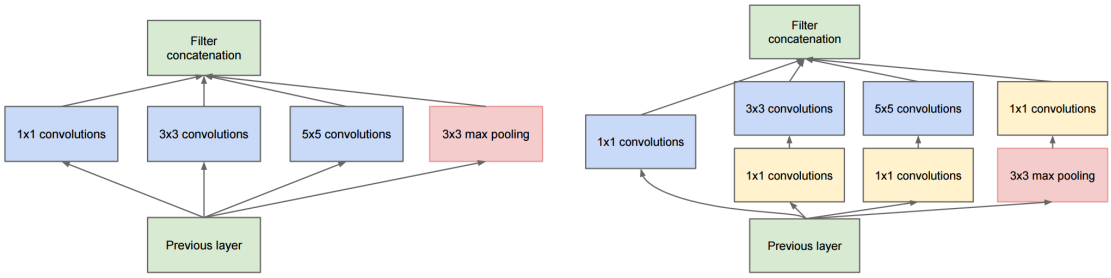
Knowing the meaning of these parameters, we can now provide a more technical definition of **transfer learning** (Pan & Yang, 2010):

- **transfer learning**: Given a source domain D_S and a learning task T_S, a target domain D_T and target task T_T, transfer learning is to help improve $f_T(\cdot)$ in D_T using the knowledge of D_S and T_S. Here $D_S \neq D_T$ and $T_S \neq T_T$.

Knowing the concepts, we can now dive into the key questions to answer in transfer learning (Pan & Yang, 2010):

- (1) What to transfer: which knowledge should we transfer? Is this knowledge relevant to the target domain/task ?
- (2) When to transfer: transfer learning should be applied to those situations where such information would improve model performance, rather than the other way round.
- (3) How to transfer: when (1) is fully discussed and evaluated, one should develop specific algorithms to apply the transfer learning. (2) should also be evaluated when developing algorithms.

In the case of supervised learning, there are two categories of transfer learning approaches in general. The first is called **inductive transfer learning**, where $D_S = D_T$ and $T_S \neq T_T$. It aims to transfer knowledge from the training task to the target task, and achieve high model performance. The other is called **transductive transfer learning**, where $D_S \neq D_T$ and $T_S = T_T$. In this situation, no labelled data would be available in the



(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 2: Inception module

FIGURE 2.29: Figure 2 of Szegedy et al. (2014), showing the (a) naive and the (b) modified ver. with dimension reduction of the 'Inception module'.

target domain, while the data in the source domain are well labelled. There are two cases of **transductive transfer learning**: $X_S \neq X_T$ or $P(X_T) \neq P(X_S)$ where $X_S = X_T$. In these cases, our approach will be to concentrate on transductive transfer learning, and we will further discuss the advantages and how we apply transfer learning in Sections 3.3.

2.8.2 Multi-Branched CNNs

The model architectures we have described so far are generally sequential: all layers are stacked together in sequence. In the case of convolutional layers, for instance, developers have to choose a customized kernel and stride size for each layer. The outputs of these convolutional layers may go into the next layer directly, or be pooled before serving as the next layer's input. A question which may then arise will be: is it possible to have a hidden layer (a) learn input features from both small and large kernels in parallel (i.e. extract image features with kernel sizes equals 1, 3 or 5 at the same time), or (b) import data inputs from the outputs of more than one previous layers, where these layers might have extracted features from the same, or different but relevant, input data (i.e. image maps of the same galaxy observed at different frequencies)? These ideas have motivated the invention of CNNs with branched modules and later Multi-branch CNNs (e.g. He et al., 2015; Szegedy et al., 2016; Li et al., 2017; Aslani et al., 2018; Zhang et al., 2019b).

Some very early and well-known branched CNNs are the GoogLeNet and later Inception networks (e.g. He et al., 2015; Szegedy et al., 2016). These networks include a structure called an 'Inception module', see Figure 2.29 of Szegedy et al. (2014). When using an inception module, the outputs from the previous layer will simultaneously go through convolutional layers with kernel sizes of 1×1, 3×3, 5×5, and a 3×3 max pooling layer in parallel. The feature maps from these layers are then concatenated together forwarded to the next layer. The architecture of an Inception module therefore is no longer in a direct top-down sequence, but has multiple branches.

Although the idea of the Inception module sounds good, such model architectures can lead to large scale outputs, requiring heavy computation power. In order to overcome these shortcomings, additional 1×1 convolutional layers were introduced before the 3×3

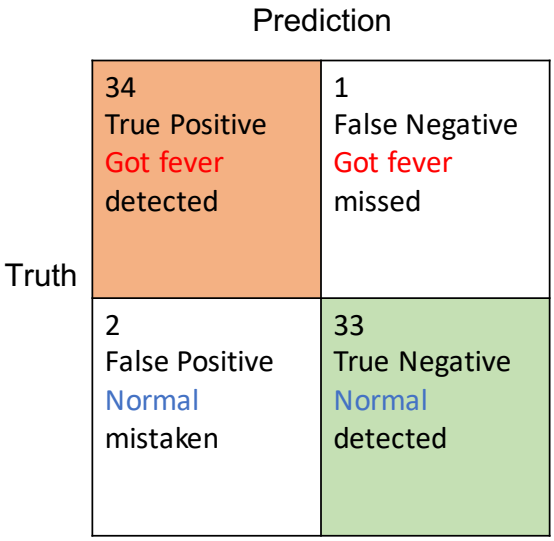


FIGURE 2.30: Illustration of a confusion matrix, showing the confusion matrix of a logistic regression model to detect fever based on the human body's temperature.

and 5×5 convolutional layers, and before the max pooling layer, which reduce the scale of the outputs.

The foundations of the 'Inception module', especially its variant ResNext (Xie et al., 2016), in some extent is similar to later Multi-branch CNNs (e.g. Li et al., 2017; Aslani et al., 2018; Zhang et al., 2019b) as they both concatenate multiple paths (Xie et al., 2016). Among the limited number of actual applications, Multi-branch CNNs can be separated into two categories: (i) those similar to the Inception Network, where a network would have a single input training sample forwarded to different paths of the CNN in parallel (e.g. Li et al., 2017; Zhang et al., 2019b), and (ii) a network that would import different but relevant input sample data that would be fed into different CNN branches and concatenated into a single output layer before being imported to the following layers in top-down sequence (e.g. Aslani et al., 2018). In actual practice, the choice of these approaches depends on the available data format and the target questions being solved, and we will discuss this further in Chapter 5.

2.9 Beyond Accuracy: Model Evaluation

So far, we have introduced FNNs, CNNs and some of their variants. When evaluating the performance of these models, we have mostly used accuracy and model loss as the key metrics. However, depending on the question we are addressing, accuracy and loss might be insufficient to evaluate a model. In this section, I will go through a number of evaluation metrics that help us to get a more comprehensive view of model performance. I will run through the concepts of Confusion matrices, Precision, Recall, and F1 score, as well as ROC curves.

To start with, a widely used tool for evaluating model performance on a class-wise basis is the confusion matrix (Stehman, 1997). It is a table to visualize model performance.

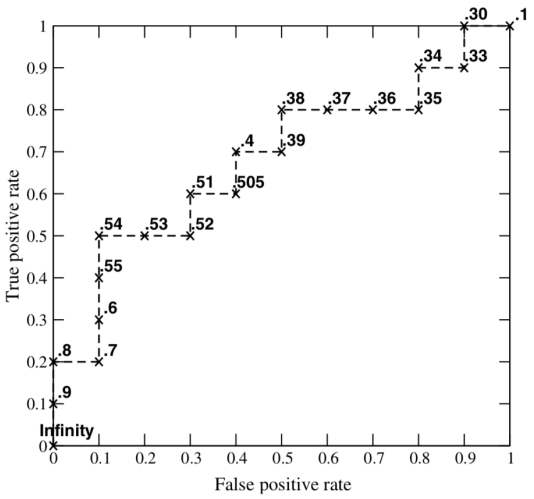


FIGURE 2.31: An illustration of ROC curve (Figure 3 of Fawcett (2006)).

Figure 2.30 gives an example confusion matrix for the fever detection problem. This matrix has four quadrants: True-Positive (TP), False-Negative (FN), True-Negative (TN), and False-Positive (FP). Assuming that in this case, the state of having a fever is "true," the matrix counts all samples from the 70 data points as TP if the model prediction matches its pre-defined class label. Similar counts are made for the other three quadrants.

The four-quadrants can be used to further assess the predictive ability of a two class model as the basis for deriving two additional metrics: **Recall** (R), **Precision** (P) (Powers, 2008). These two metrics are calculated as:

$$\mathbf{P} = \frac{TP}{TP + FN} = success\ rate\ of\ finding\ people\ that\ have\ a\ fever. \tag{2.64}$$

and

$$\mathbf{R} = \frac{TP}{TP + FP} = sensitivity\ level\ of\ the\ model\ when\ detecting\ fever. \tag{2.65}$$

In the context of fever detection, both **R** and **P** are both important as we neither want to miss a patient that needs necessary treatment, nor want to mistakenly identify people with normal body temperature as it will leads to unnecessary treatment and might have side effects. Accordingly, if one wishes to evaluate the model performance with a single metric, then the F_1 score might help:

$$\mathbf{F1} = 2\frac{\mathbf{P} \times \mathbf{R}}{\mathbf{P} + \mathbf{R}}. \tag{2.66}$$

In short, the **F1 score** is the weighted average of **R** and **P**, providing a general assessment when identifying samples of a class. In this example, the model has a **R** = 0.94, **P** = 0.97 and a **F1 score** of 0.96.

Besides the three metrics, the Receiver Operating Characteristic (ROC) curve is also a nice tool to represent model performance (Fawcett, 2006). The ROC curve is most often used for binary classification problems, although it potentially can be used for multi-class logistic regression problems as well. The ROC curve is a visualisation of false-positive

rate versus true-positive rate for sample candidate thresholds from 0 to 1. The true-positive rate is equal to the recall, **R**, when the considered class is the 'real' class. The False-positive rate, on the other hand, is defined as:

$$False\ positive\ rate = fpr = \frac{\mathtt{FP}}{\mathtt{FP} + \mathtt{TN}}. \tag{2.67}$$

Considering the aforementioned confusion matrix, the **fpr** of the example model refers to the rate at which a model mistakenly identifies those who have normal body temperatures as having a fever. The ROC curve can be seen as a trade-off between the two variables, and the area under the curve (AUC) value is often used when comparing models. In such a comparison, testing on the same samples, the model with a higher AUC is considered to perform better in distinguishing people with or without fever correctly. The application of these metrics/curves shall be further discussed in Chapter 3 and 5.

2.10 FNNs/CNNs in Astronomy: A Brief Review

In this section, we will briefly run through the FNN/CNN applications in astronomy between 1980s and 2010s. Considering the work being done in this thesis, we here only discuss traditional supervised FNN/CNN applications, with no other machine learning approaches involved.

2.10.1 1980s-1990s: MLP with back-propagation

Dated back to 1980s, back-propagation was announced (Rumelhart et al., 1986). Soon after that, early supervised learning applications using neural network in astronomy then were implemented in both detector front-end and data analysis (Miller, 1993), when most applications used MLP/FNN.

Typically, applications at the time have covered fields such as adaptive telescope optics, object classification and matches, detector event filtering, and have also been applied on satellite systems (Miller, 1993). We here provide two examples: wave front correction and point source classification.

The power of neural network applications for wave front correction was firstly investigated upon the Multi-Mirror Telescope, a telescope with 6 1.8 m mirrors (Angel et al., 1990). An FNN was built and trained with simulated images, returning the predicted piston position and (x, y) tile for each mirror. When the network was tested upon 500 test images, it reached a resolution of $0.06''$, close to the diffraction limit of MMT.

Early attempts have also focused on object classification. Adorf & Meurs (1988) as an example, pioneerly used primary FNNs when classifying objects in the IRAS point source catalogue (Adorf & Meurs, 1988; Murtagh, 1988). Though the team have also applied other methods such as Principal Component Analysis (PCA; Jolliffe & Cadima, 2016), they claimed it is feasible to use neural networks for their task (Adorf & Meurs, 1988). Attempts in using neural networks have also been done upon fields like star/galaxy classification (Odewahn et al., 1992), galaxy types identification (Storrie-Lombardi et al.,

1992), cosmic ray detection (Murtagh & Adorf, 1992). However, due to the limit of available training data with good quality, at the time it was believed at the time that neural networks are not ideal for object classification (Miller, 1993). A complex network trained with small amount of data may have a higher risk of over-fitting.

2.10.2 2000s: Early involvement of image inputs

Entering the 21^{st} century, most applications in the early 2000s were still focused on FNNs, and these have been widely applied in research fields such as galaxy classification and spectral classification. These approaches did not only consider pre-selected features (Bazell & Aha, 2001; Bailer-Jones, 2000; Banerji et al., 2010), but also start to import pre-processed images directly (Goderya & Lolling, 2002).

As a pilot project in the field, the Goderya team selected 250 galaxy images from the Digitized Sky Survey at the Space Telescope Science Institute web page as a training/-validation dataset. They then pre-processd and resized these images to 30×30 pixels as inputs, and used them as the input to an FNN. The network aimed to identify the morphology of elliptical, simple-spiral and barred-spiral galaxies. Sadly, due to the relatively small data sample and large number of neurons, the network only correctly identified 19 out of 37 validation samples (Goderya & Lolling, 2002).

2.10.3 2010s: Growth of CNNs

Although simple CNNs such as LeNet-5 had been released in 1998, the growth of relevant astronomical applications did not appear until AlexNet won the 2012 ImageNet challenge (Krizhevsky et al., 2012a). ImageNet is a image-based multi-class data set with millions of manually labelled images. Every year the ImageNet project would hold an annual competition called the *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC), where developers are called to build classifiers and identify a subset of the ImageNet dataset. The rise of these algorithms triggered the 'deep learning revolution' (Venn et al., 2019). By 2014, all the top ImageNet algorithms were using deep architectures.

Given that the launch of simple and successful CNNs like LeNet-5 could be traced back to 1998, the broad application of CNNs in recent years has become possible thanks to the development of proper parameter initialization (Glorot & Bengio, 2010), advanced activation functions (e.g. ReLU; Hahnloser et al., 2000), optimisers with better performance (e.g. Adam; Kingma & Ba, 2014), as well as the availability of high performance Graphic Processing Units (GPUs). These tools, back in the days when LeNet-5 launched, were not available (Venn et al., 2019).

As the necessary tools became accessible, machine learning was increasingly applied in astronomy research over the last decade. Large-scale ML exploration for astronomical studies started around 2016, and with over 300 refereed papers in 2019 (see Figure 2.32; Venn et al., 2019).

To date, the applications of neural networks, or machine learning in a broader sense, have covered a large number of fields such as gravitational lens finding (e.g. Metcalf

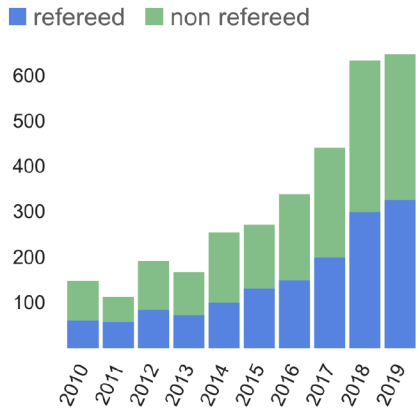


Figure 1: Astronomy papers that include machine learning methods in the abstract or title since 2010. From the Astrophysics Data System (ADS).

FIGURE 2.32: The count of annual ML applications in the field of astronomy research (Figure 1 of Venn et al. (2019)).

et al., 2019), supernova finding (e.g. Moss, 2018), galaxy merger detection (e.g. Ackermann et al., 2018), optical galaxy morphology classification (e.g. Dieleman et al., 2015), and radio galaxy morphology classification (e.g. Aniyan & Thorat, 2017).

In order to have a closer look, galaxy morphology classification could be a good example. A very early CNN application in the field was made by the Galaxy Zoo project, where they attempted to use state-of-the-art CNNs at the time to identify optical/infrared galaxy images, and won the Galaxy Challenge (Dieleman et al., 2015). Soon after, Huertas-Company et al. (2015) extended the method to around 50 000 galaxies with high redshifts. A few years later, CNN based classifiers to identify radio galaxy morphologies were developed as well (e.g. Aniyan & Thorat, 2017; Pourrahmani et al., 2018; Lukic et al., 2019; Tang et al., 2019).

Another example is the gravitational lens hunting (e.g. Lanusse et al., 2017; Jacobs et al., 2017; Schaefer et al., 2018; Petrillo et al., 2019). A few CNN approaches were included in the strong gravitational lens finding challenge, largely to speed up the finding process, and achieved human-comparable (or better) identification accuracy (e.g. Schaefer et al., 2018; Venn et al., 2019).

Chapter 3

Radio Galaxy Classification using Convolutional Neural Networks

The work in this chapter is published in Tang, Scaife & Leahy, 2019, *Monthly Notices of the Royal Astronomical Society*, Volume 488, Issue 3, Pages 3358–3375.

Traditionally, identifying FR class is done by visual inspection and astronomers would find hot spot peaks of an extended radio source and measure the angular distance between them. Later, they would estimate the edges of the source, and measure their source extent. The ratio of the two distances would then lead to an FR class identification result of the source. Sometimes, classification would also consider source jet angle and other morphological characteristics. Such an approach has been widely used over the past few decades and has achieved great success (e.g., Subrahmanyan et al., 1996; Cotter et al., 1996; Leahy et al., 1996; Ishwara-Chandra & Saikia, 1999; Saripalli et al., 2005; Machalski et al., 2007; Solovyov & Verkhodanov, 2011; Kuźmicz & Jamrozy, 2012; Butenko & Tyul'bashev, 2016; Dabhade et al., 2017). However, I note that this approach can hardly deal with radio catalogues produced by next generation of radio surveys such as EMU.

The anticipated volume of objects from new radio surveys have motivated the introduction of semi-automatic and automatic object classification algorithms. Recently, Convolutional Neural Networks (CNN; Krizhevsky et al., 2012b) were applied to radio galaxy morphology classification (e.g. Aniyan & Thorat, 2017; Ma et al., 2019). These studies suggest that complex radio source structures can be identified and classified according to their morphology from images drawn from a single survey. Nevertheless, it is still unclear whether a model trained using one survey is transferable to other survey data. How to build a dataset cost-efficiently and maximize the generalisation of machine learning models across multiple surveys remain open questions. One solution to these two questions, however, might be transfer learning (Yosinski et al., 2014).

In this chapter, I investigate the applicability of transfer learning to the classification of radio galaxy morphology using survey data. In Section 3.1 I describe the construction of our training, test and validation data sets, including data acquisition, image preprocessing and data formatting. I introduce the model architecture used for classification in Section 3.2 and in Section 3.3 I describe the practice of transfer learning in the context

of the radio astronomy domain. In Section 3.4, I compare and discuss the performance of these strategies, and discuss the applicability of our results to future radio surveys.

In what follows I assume a ΛCDM cosmology with $\Omega_m = 0.3153$, $\Omega_\Lambda = 0.6847$, and a Hubble constant of $H_0 = 67.36$ km s^{-1} Mpc^{-1} (Planck Collaboration et al., 2018). Computational work in this chapter was done using a system with 32 Intel Xeon(R) E5-2640 v3 CPUs at 2.6 GHz and 202.5 GB memory.

3.1 Construction of the Training Set

The data sample used in this work is designed to train a machine learning model capable of automatically classifying FR I and FR II radio galaxies. I use input data extracted from the NVSS (Condon et al., 1998) and FIRST (Becker et al., 1995) radio surveys. Here I describe the data selection process as well as the automated acquisition methods used.

3.1.1 Astroquery based image data batch download

I selected an input sample of radio galaxies following a similar methodology to Aniyan & Thorat (2017). This method uses the Combined NVSS and FIRST Galaxies catalogues (CoNFIG; Gendre & Wall, 2008; Gendre et al., 2010), and the FRICAT catalogue (Capetti et al., 2017a) to select objects from the FIRST survey. These catalogues were selected as they share significant source populations, the data are freely accessible, and the sources are well-resolved in the FIRST images (Aniyan & Thorat, 2017).

The CoNFIG catalogue contains 859 resolved sources divided into 4 subsamples (CoNFIG 1, 2, 3 & 4) with flux density limits of $S_{1.4\,\mathrm{GHz}} \geq 1.3, 0.8, 0.2, 0.05$ Jy, respectively. These sources are selected from the NVSS survey within the northern field of the FIRST survey.

Redshifts for the catalogue sources were obtained either from SIMBAD, the NASA Extragalactic Database (NED) or estimated by using the $K_s - z$ relationship (Gendre et al., 2010) with source K_s magnitudes from the 2MASS survey (Skrutskie et al., 2006) where available. From the full sample, 638 sources were associated with redshifts.

Source morphologies in the CoNFIG catalogue were manually identified by looking at their NVSS and FIRST contour maps. Objects were classified as FR I, FR II, *compact*, or *uncertain* (Gendre et al., 2010). Sources with collimated jets, showing hotspots close to their cores, were classified as FR I; sources with their lobes aligned and hotspots situated at the edges of the lobes were classified as FR II. If the source morphology was ambiguous, it was classified as *uncertain*. Sources with sizes smaller than 1″ were classified as *compact*. In all, 95.7% of CoNFIG sources have their radio morphology classified. I note that the FR I sample in this catalogue also includes sources that are considered to be wide-angle tail or irregular by other studies (Leahy, 1993).

Depending on whether a candidate in the CoNFIG sample showed a clear FR I or FR II morphology, the identification of each object was qualified as 'confirmed' (c) or 'possible' (p). 'Confirmed' sources were confirmed using the VLBA Calibrator Surveys (Beasley et al., 2002; Fomalont et al., 2003; Petrov et al., 2006; Kovalev et al., 2007) or the

Pearson-Readhead survey (Pearson & Readhead, 1988). The final catalogue contains 50 confirmed FR I objects and, 390 confirmed FR II sources.

To balance FR I and FR II sample sizes, the FRICAT catalogue of FR I radio galaxies was introduced. This catalogue, which contains 219 FR I radio sources, is a subsample of the catalogue from Best & Heckman (2012), hereafter BH12. BH12 was formed by cross-matching the optical spectroscopy catalogues produced by Brinchmann et al. (2004); Tremonti et al. (2004) based on data release 7 of the Sloan Digital Sky Survey (SDSS DR7; Abazajian et al., 2009) with the NVSS and FIRST surveys, for sources with flux densities in NVSS that were greater than 5 mJy.

To make the FRICAT sample, the authors compiled all BH12 sources with $z < 0.15$, resulting in 3,357 objects. From this sample, all sources with radio emission extended more than 30 kpc from the host galaxy center were selected, resulting in a sample of 741 sources with well-resolved morphologies. From these 741 sources, individual objects were then classified as FR I type if (1) a one-sided or two-sided jet was present (including sources with bent jets), (2) the surface brightness of the jet decreased along its length, and (3) there was little or no brightness enhancement at the end of jet.

Sources in which brightening was observed along the jet (Burns et al., 1982), were excluded. All three FRICAT authors classified the sources independently, and an object would be added to the catalogue only if 2 out of 3 authors agreed. The final catalogue contains 219 FR I radio galaxy candidates. The hosts of the FRICAT sources were found to be luminous ($-24 \geq M_r \geq -21$) early-type galaxies with black hole masses in the range 10^8 to 3×10^9 $M_\odot$.

CoNFIG and FRICAT together form the input sample for this work, where I only consider the samples with confident FR morphology identification. I did not include the sample objects with uncertain morphology due to the nature of classical CNN: sample labels imported to a classical CNN are believed to be the 'ground-truth', giving a certain prediction probability of either 0 or 1. One would then train a CNN model to make probabilistic prediction as close to the 'ground-truth' as possible. It is then less practical to label uncertain samples with 0 or 1 probabilities.

By cross-matching FR I source centroid coordinates from FRICAT and CoNFIG, I found 3 duplicate sources. To remain consistent with Aniyan & Thorat (2017), I did not remove this small number of duplicate objects. The final sample comprises 266 FR I sources and 390 FR II sources. I extracted the central coordinates of the host galaxy in each case and used these to download FIRST images for each object using the python `astroquery` and `urllib` tools.

The `astroquery` library is an affiliated package of `astropy`, with tools for querying astronomical web forms and databases. I used the `astroquery.skyview.get_image_list` function, specifying central source coordinates, setting survey name to `'VLA FIRST (1.4 GHz)'` or `'NVSS'`, and specifying the image scaling to be `'Linear'`. With the returned list of image URLs I then used the `urllib.request.urlretrieve` function to download NVSS and FIRST images. `SkyView` by default returns each FITS image with size of 300 × 300 pixels

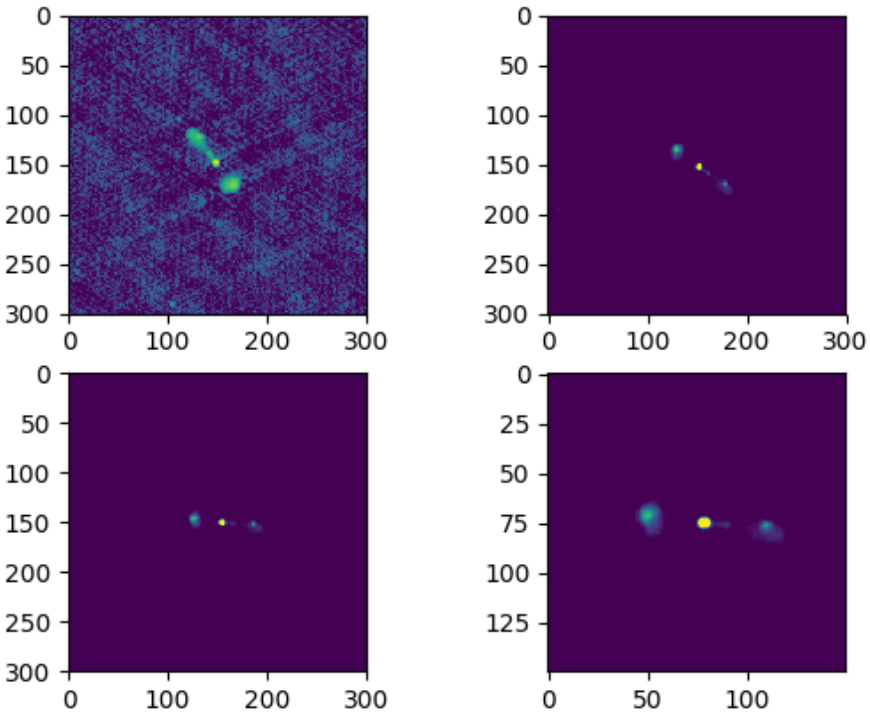


FIGURE 3.1: An example of image pre-processing and augmentation. The upper left image is the log scaled original image downloaded from SkyView. The other three images, from left to right, top to bottom are the ones experienced sigma-clipping, rotation, and centered crop. The radio source centered at the sample FIRST image is 4C 31.30, a 'confirmed' CoNFIG FR II sample. The radio galaxy host locates at (J2000) 07:45:42.13 +31:42:52.6.

(0.15×0.15 degrees for FIRST images). The pixel values in these FITS images corresponds to a brightness scale in units of Jy/beam.

3.1.2 Image pre-processing and augmentation steps

The image pre-processing in this work consists of three operations: pixel-value re-scaling, image rotation and image clipping. These are performed on all input images.

Aniyan & Thorat (2017) reported that image background noise decreased classifier performance. Having investigated various noise clipping options, they proposed a solution where pixel values lower than 3 times the local rms noise were set to zero, which I followed.

After clipping, I re-scaled each image following:

$$\text{Output} = \frac{\text{Input}-\text{Min}}{\text{Max}-\text{Min}} \times (255.0 - 0.0)$$

where `Max` and `Min` refer to the maximum and minimum pixel value in an image. `Output` and `Input` represent the final re-scaled and original pixel values in the image, respectively. Re-scaled images have pixel values in the range from 0 to 255. These are then

Class	Training/Validation	
	Original	Augmented
FR I	189	13,797
FR II	273	13,650
Total	462	27,447
Class	**Test**	
	Original	**Augmented**
FR I	80	5,840
FR II	117	5,850
Total	197	11,690

TABLE 3.1: A summary of FR I, FR II images of the dataset samples. The dataset consists of samples for training, validation, and testing. The augmented samples are created by the process claimed in Figure 3.1. In step of 1°, FR I source images were rotated from 1° to 73°. For FR II images, I rotated them from 1° to 50°.

saved in PNG format.

When training machine learning models, training datasets typically have sizes of order ∼10,000 data samples (Aniyan & Thorat, 2017). Here, the original dataset contains a total of 659 source images, where ∼30% will serve as test samples to evaluate our trained model performance (Aniyan & Thorat, 2017). Data augmentation was therefore considered to be necessary.

Considering both dataset size and class balance, I decided to augment our dataset to have around 14,000 training images and 6,000 test images for each class. Following previous practice (Aniyan & Thorat, 2017), data augmentation was done by rotation of the original input source images by 1°, 2°, 3° etc. Table 3.1 provides details of our dataset sample size. The final dataset contains 39,796 sample images.

The final step in building our dataset was to clip the image sizes. Image clipping is to designed to constrain an image to its central source, yet remain large enough to identify structure. Aniyan & Thorat (2017) clipped sample images from their centers to 150 × 150 pixels, which is equivalent to a physical extent of 274.1 kpc at $z = 0.05$ for FIRST images. For NVSS images at the same redshift, however, image physical extent equals to 2283.8 kpc. For the CoNFIG and FRICAT samples, 96.4% of objects have redshifts ≤ 0.05. I visually inspected the central source of sample images from our dataset clipped in the same way, and found that the image width was suitable to recognize the sources while retaining characteristics sufficient to identify their FR morphology. Given that the sky coverage of NVSS sample images is much larger than that of FIRST images, these images would inevitably include secondary sources. The effect of this angular extent difference will be discussed in Section 3.2.

3.1.3 Data formatting and division

After pre-processing, I converted our input dataset images into `numpy` format. Target classification labels were defined as one-hot vectors: FR I samples are labeled as [1., 0.],

while FR II samples have the label [0., 1.]. (In Section 3.2 I will explain why such label vectors are convenient to use and will simplify the computational process when training the model.) Label and image datasets were saved in 2-dimensional arrays. Rather than saving images in 3-dimensional arrays, I saved data input in 4-dimensional arrays as they are more flexible, capable of saving both single channel images (greyscale) and multiple channel images (e.g. RGB), and share the same format as the well-known MNIST dataset (Lecun et al., 1998b). I also note that data were fully shuffled before being imported into the CNN.

I split the master dataset into training, validation, and test subsets. Each subset of data includes both image data and target classification labels. The training set is used to train the CNN machine learning model via back propagation. The validation set, on the other hand, helps us to examine whether the model is over-fitting the data through forward propagation. This examination takes place at every training epoch. Validation subset samples can therefore be extracted from the original training set. The test data subset is separated from the training set and validation set, with samples unseen by model before testing. This subset provides samples for evaluating the performance of the trained model when doing realistic classification. Table 3.1 gives an overview of the data subsets used in this work.

In this work, consistent with Aniyan & Thorat (2017), I separated training and test samples with a ratio of 70-30. I then pre-processed and applied image augmentation on all sample images. The primary training set is split into training and validation using a ratio of 80:20.

3.2 Network Architecture

An appropriate network architecture should consider object complexity and computational power. If necessary, transfer learning ability is a factor as well. Although simple networks may perform well for some applications (Gheller et al., 2018), classifying radio galaxy morphology requires comparatively deeper networks. Early attempts have shown that, for radio galaxy classification, simple networks perform only slightly better than random guesses (Aniyan & Thorat, 2017). With fewer learnable layers, simple networks may also meet issues when using transfer learning as they have weaker expressive ability (Oquab et al., 2014). A reasonably deep network therefore becomes necessary for classifying radio galaxies and being capable of doing so using transfer learning.

Among the pre-existing architectures, the AlexNet CNN is a representative deep network (Krizhevsky et al., 2012b). This is a widely used 12-layer parallel computing CNN with 5 convolutional layers, 3 max pooling layers, 3 fully-connected layers, and a softmax readout layer. Aniyan & Thorat (2017) slightly modified this network and succeeded in classifying FR I, FR II, and Bent-tailed radio sources, with a general accuracy of above 90%. Their network used a GPU-based implementation and was able to train 30 epochs

of data in ∼70 mins (Aniyan & Thorat, 2017). The number of epochs in machine learning refers to the number of times that the complete training dataset is imported during training, where in this work the complete training dataset includes augmentations.

Inspired by Aniyan & Thorat (2017), I adopt a similar but simplified 13-layer network see Figure 3.2. I initially discard the network components allowing for parallel computation. I then added another fully connected layer to reduce over-fitting. Finally, instead of optimizing loss using a traditional mini-batch gradient descent optimizer (Robbins & Monro, 1951) and step decay learning rate schedule, I used the adaptive mini-batch optimizer AdaGrad (Duchi et al., 2011) to minimize the model loss function. This optimizer algorithm works with a batch size of 100 and a initial learning rate of 0.01. The batch size refers to the number of training samples fed into the network at a time. Since I use 22,268 training samples, I import our data in 223 batches. Such a data import method is often referred to as mini-batching. The mini-batching method has typically been found to speed up the training process (Benatan & Pyzer-Knapp, 2018).

Using this architecture, I observed that model validation accuracy started to saturate after 10 epochs (Aniyan & Thorat, 2017). Consequently, all training in this work is stopped after 10 epochs. The filter:node number for each layer is set to be 1:16 due to computational power limitations. The same architecture is used for both our initial model training and later transfer learning applications. Table 3.2 provides further network details. Notably, the `parameters` characteristic for each layer in Table 3.2 is determined by the receptive field size, the input channel number, and the depth of each layer. The `Conv1` layer, for instance, has $11 \times 11 \times 1 \times 6 + 6 = 732$ learnable parameters.

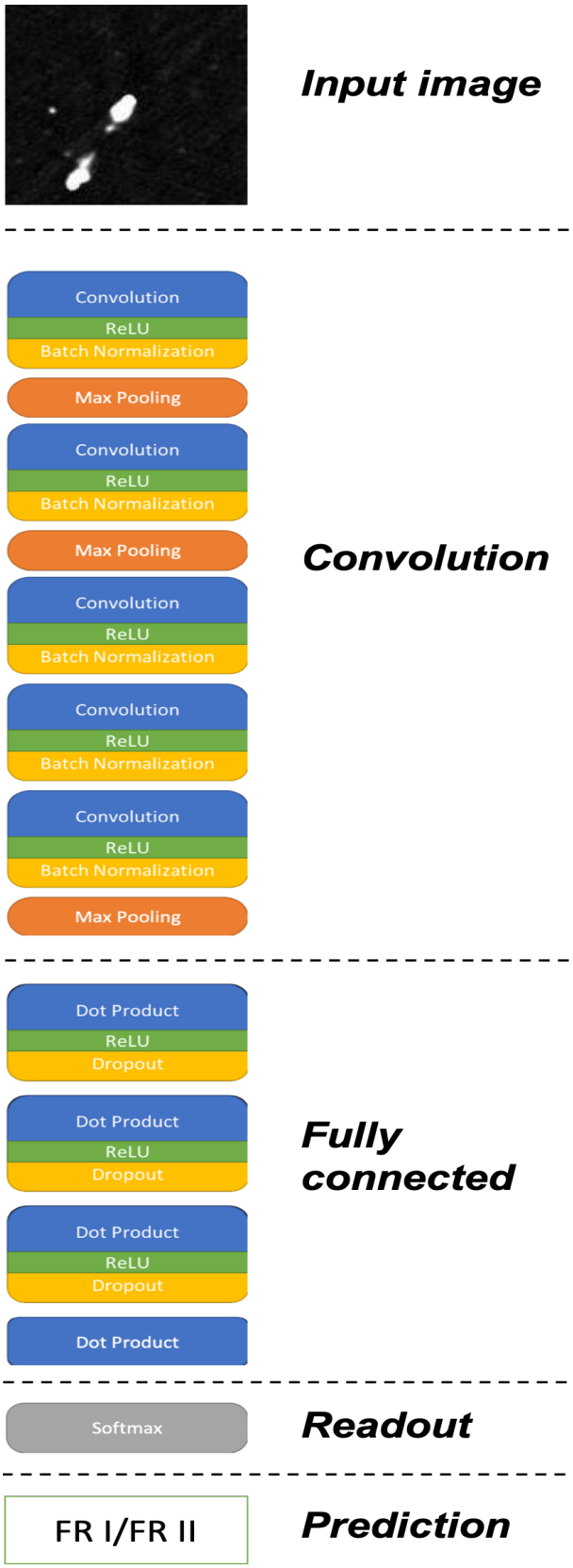


FIGURE 3.2: Network architecture adopted in the work. Blue: filters with learnable parameters; Green: activation functions; Yellow: Regularizers; Orange: Pooling layers; Grey: Softmax layer. The 13-layer architecture contains 5 convolutional layers, 3 max-pooling layers, 4 fully-connected layers, and a softmax readout layer. I consider pooling and readout layers separately.

Layer	Name	Receptive Field	Stride	Input Channel Number	Depth	Activation	Regularizer	Parameters
1	Conv 1	11×11	1	1	6	ReLU	Batch Normalization	732
2	Max Pooling 1	2×2	2	6				
3	Conv 2	5×5	1	6	16	ReLU	Batch Normalization	2,416
4	Max Pooling 2	3×3	3	16				
5	Conv 3	3×3	1	16	24	ReLU	Batch Normalization	3,456
6	Conv 4	3×3	1	24	24	ReLU	Batch Normalization	5,184
7	Conv 5	3×3	1	24	16	ReLU	Batch Normalization	3,456
8	Max Pooling 3	5×5	5					
9	Fully Connected 1		1	$5 \times 5 \times 16$	256	ReLU	Dropout	102,656
10	Fully Connected 2			256	256	ReLU	Dropout	65,792
11	Fully Connected 3			256	256	ReLU	Dropout	65,792
12	Fully Connected 4			256	2			514
13	Loss Softmax							
Total Parameters:								249,998

TABLE 3.2: Network parameters of the classifier I adopted in the work. 'Parameters' are only available for 'Conv' and FC layers. Parameters within these layer can learn through back propagation, while pooling and loss layers cannot learn.

3.2.1 Direct Classification

CNNs are capable of learning the form of the features for classification and expressing them as the values of the convolution kernel weight matrix. In the context of radio morphology classification, CNNs will learn these features from the input training samples and extract differentiable features from them. Figure 3.3 shows some typical examples of training samples.

When a convolutional network trains on these samples, its initial convolutional layers will tend to learn general sample features, while lower layers are more likely to extract features specific to the dataset itself. I visualize these features by plotting feature maps, which show the activation of different parts of the image (Zeiler & Fergus, 2013). Figure 3.4 shows feature maps for two representative samples. Features in the diagram were extracted from a randomly initialized model following 10-epoch training. These features correspond to the 2nd and the 5th convolutional layers. Both layers implement 16 filters in total and the figure shows the first 10. Generally speaking, for the FR II source (bottom), the features learned by the 2nd layer seemed to emphasize the existence of double edge-brightened lobes and the relative positions of the hotspots. Whereas the lower layers have learned more specific features: source outlines or the source-background relationship. This is similar to what was observed in Aniyan & Thorat (2017). All these features are saved via their model weights, and are used by the fully connected layers to make an FR binary classification.

Both NVSS and FIRST training samples were imported to train a randomly initialized model for 10 epochs each. Figure 3.5 provides an overview of model learning processes. The average temporary accuracy and loss is calculated from 10 independently trained models. The standard deviation of these parameters corresponds to their error bars in the diagram. All models trained from random initializations using the Xavier uniform weight initializer (Glorot & Bengio, 2010) that experienced 10 epochs of training have a validation accuracy above 99.5% and losses lower than 0.02, regardless of input sample selection.

Models trained using FIRST data as the input samples tended to have a smaller error. These models also learned more efficiently, as their training losses dropped faster. Models trained using NVSS images as input samples seemed to oscillate more frequently, which implies that these models have a higher risk of making random guesses and a lower chance of performing stable training.

Reasons for this oscillation could be the extent of the NVSS sample images. I trained and tested on the same NVSS sample images but extracting only the central 18×18 pixels, which share the same sky area as the FIRST sample images. The network architecture was consistent with that previously used, with the exception that that no pooling layer is used during test. Qualitatively, the level of the oscillation decreases in this case. This is unsurprising as our original NVSS sample images contain background sources, which might disturb the network when extracting class features. Also, the comparatively poor resolution and sensitivity of NVSS versus FIRST could cause relatively large error and strong oscillation. Some other possible causes are discussed in Section 3.3

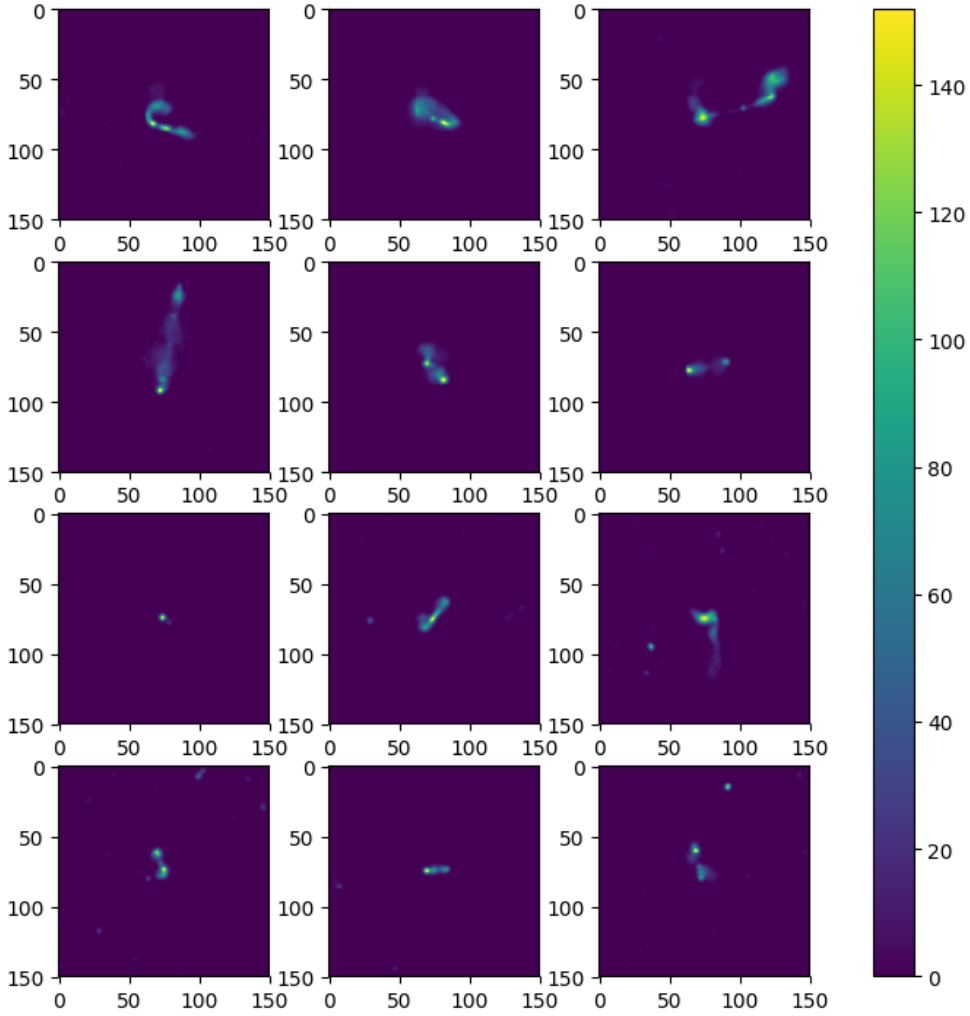


FIGURE 3.3: Examples of images used in model training. Models trained with these samples were used to classify FR morphology from test dataset NVSS or FIRST images. 1st row: FR I samples of FIRST images; 2nd row: FR II samples of FIRST images; 3rd row: FR I samples of NVSS images; 4th row: FR II samples of NVSS images. The color bar represents the linear-normalized pixel values.

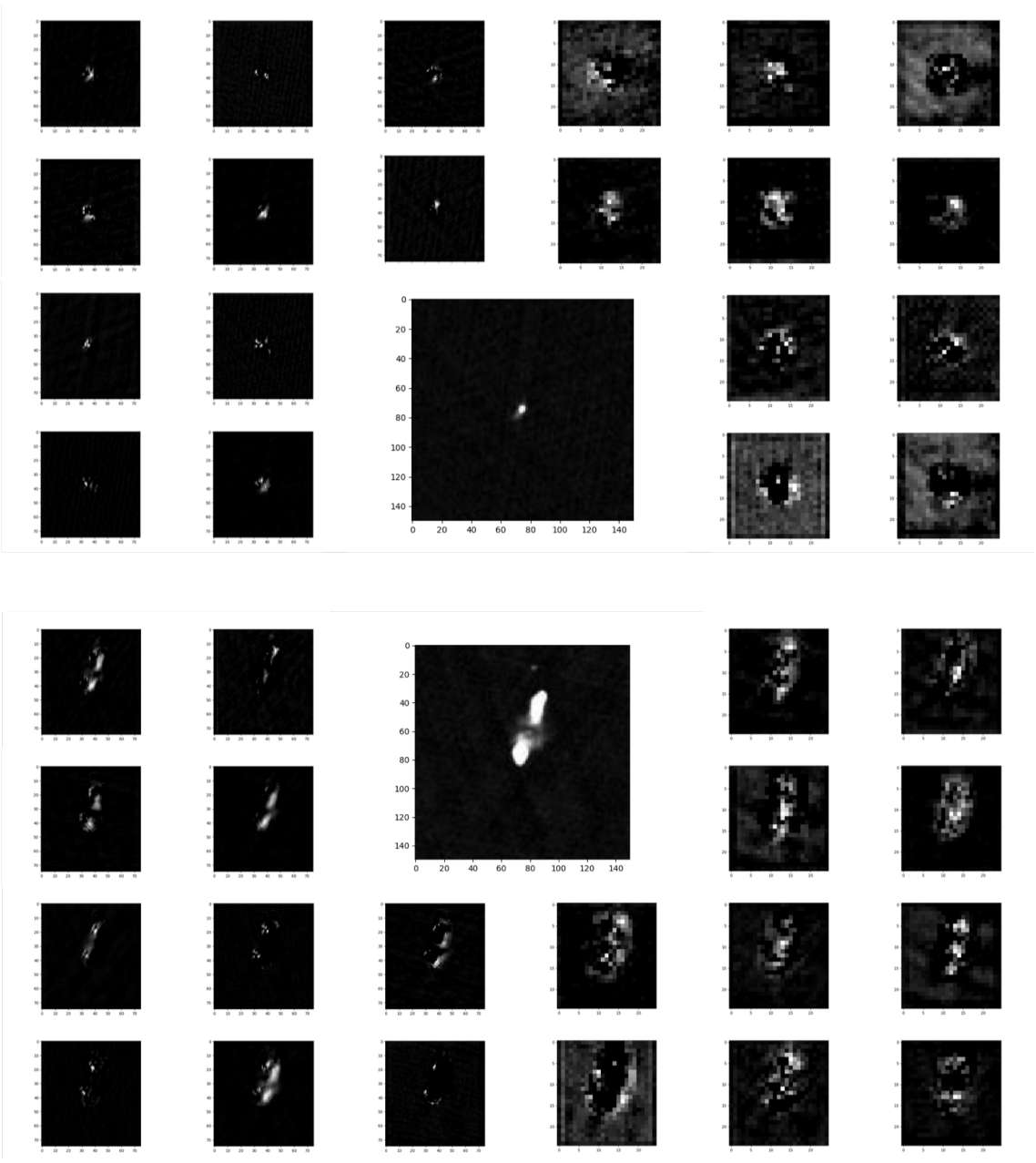


FIGURE 3.4: An example of feature maps using testing FIRST sample images. These images are produced by convolving the example source image with the first 10 filters of either the second or the fifth convolutional layer shown in Figure 3.2. Upper-middle: An example of FR I sources in the testing set. Lower-middle: An example of FR II sources in the testing set. Upper-left: Features of the example FR I source extracted by the second convolutional layer. Upper-right: Features of the example FR I source extracted by the fifth convolutional layer. Lower-left: Features of the example FR II source extracted by the second convolutional layer. Lower-right: Features of the example FR II source extracted by the fifth convolutional layer. Source images and feature maps in the diagram are in grayscale.

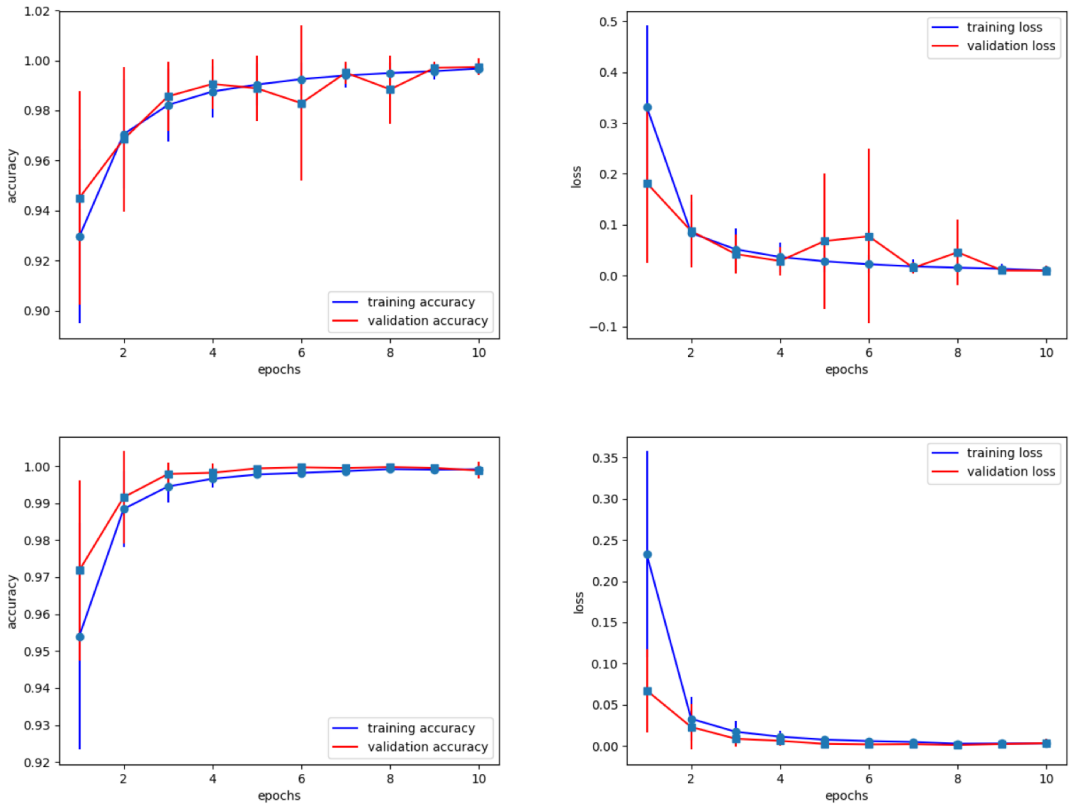


FIGURE 3.5: Upper: averaging learning and loss curve for 'Xavier' models trained on NVSS images. Lower: The same curves for models trained on FIRST images.

3.3 Transfer Learning

Constrained by the lack of large-scale labelled astronomical training data, as well as limited computational power, recent astronomical applications have turned their attention to pre-trained models (Wu et al., 2019). These models were trained with ImageNet, a sizeable visual dataset with a thousand object categories and millions of labeled sample images from daily life (Russakovsky et al., 2014). As expected, the performance of these generalized models when applied to astronomical survey data was found to be inferior to that of custom models for direct classification (Lukic et al., 2018).

An alternative to direct classification using inherited weights is to initialize from pre-trained model weights before performing customized training. One can restore model weights for either just the initial (Wu et al., 2019; Domínguez Sánchez et al., 2019) or all layers in a network (Ackermann et al., 2018; Domínguez Sánchez et al., 2019). In some cases, such an approach has been shown to help customized models to avoid over-fitting and/or to help them to accelerate the training process (Ackermann et al., 2018; Wu et al., 2019).

Known as *transfer learning*, this approach can re-use knowledge from a solved problem and apply it to relevant but different applications (Pratt, 1993). For instance, learning general features from handwritten digits and applying them in handwritten character recognition (Maitra et al., 2015). In the context of classifying FR morphology in radio galaxies with CNNs, generic features learned from initial network layers, such as source edges and hot spots, should exist irrespective of the radio survey; complex features, however, would be learned by the last few layers (Aniyan & Thorat, 2017). Using this approach to mitigate against model over-fitting during the training process is referred to as regularization (Ackermann et al., 2018).

The application of transfer learning for classification requires careful consideration of which layers to train (Sonntag et al., 2017). When applying transfer learning, a network is trained on a new dataset of samples using the stored weights for some or all layers from a pre-trained network as an initialization, rather than initializing weights randomly. The choice of which layers to restore depends on the size of the dataset used to train the pre-trained model, the size of the new dataset, the correspondence of the two datasets, and the architecture of the pre-trained network.

Here, I consider the use of pre-trained models trained on radio survey data. Rather than learning everyday object features, layers of the CNN learn radio galaxy features such as jets and hot spot relative positions. These features are universal in classifying source FR morphology, irrespective of survey.

The application of transfer learning has two major advantages. Firstly, by inheriting general morphological features from a pre-trained model, a new model may have a better starting point. Secondly, freezing the weights for the convolutional layers can significantly reduce the training time required to achieve comparable accuracy. However, although loading weights from pre-trained models is often practical, the method needs to be treated carefully. High-level features needed to be trained on customized datasets to

learn features for a specific classification problem. In addition, as a pre-trained model and customized model are usually addressing different objectives, transfer learning might produce a `NaN` loss function value (Wu et al., 2019).

Furthermore, transfer learning strategies vary depending on network architecture and dataset definitions (Aniyan & Thorat, 2017). Inappropriate architectures can lead to over-fitting, which can become severe when the re-trained model uses only a small number of new training samples but has a considerable number of learnable parameters (Wu et al., 2019). The relative training sample size for pre-trained and transfer-learning models requires careful consideration in order to obtain good model performance (Domínguez Sánchez et al., 2019).

In the context of classifying radio morphology, the influence of transfer learning using data from surveys at different frequencies and with different resolution remains unclear. Here, I explore this question using a transfer learning approach implemented on a variation of the AlexNet CNN (Krizhevsky et al., 2012b; Aniyan & Thorat, 2017). As part of this work:

- I develop a quasi-automatic pipeline to construct training datasets from archival radio surveys. This can be used to download and process images from various surveys. This pipeline makes comparing models trained on different surveys possible.
- I convert the data samples to a dataset format consistent with the standard MNIST machine learning dataset. This enables the datasets to be used in other network architectures.
- I simplify and modify the AlexNet CNN architecture, a widely accepted CNN architecture recently adopted in radio galaxy classification. The primary network requires parallel GPU computation, which is not considered in this work. Our resulting network can be trained and tested with modest computation power to provide end-to-end training and classification.
- I develop and implement three different transfer learning strategies. Pre-trained models are trained on either NVSS or FIRST images, with final transfer learning models transfer-learned on the other.
- I demonstrate that the architecture can be used to classify FR radio morphology and achieve accuracy comparable with the performance of human radio astronomers.
- I evaluate the feasibility, training time cost, and model performance when applying different transfer learning strategies. I examine the possibility of using the same classifier to make prediction on both NVSS and FIRST images.

This work provides an alternative method to full network training for radio galaxy classification and demonstrates under what circumstances such an approach is valid.

3.3.1 Training Strategies

In practice, transfer learning can be done by freezing the initial layers and re-train only the last layers (Aniyan & Thorat, 2017). However, it is uncertain if this can be valid regardless of the choice of which input survey is taken as a starting point and which layers are frozen. Here I considered three transfer learning strategies:

- **Method 0:** Inherit the complete network architecture and weights from the pre-trained models and re-train the full network.
- **Method A:** Inherit the same network and weights for all layers and re-train the fully connected layers.
- **Method B:** Inherit the same network and the weights for the convolutional layers, but re-train the fully connected layers from scratch.

Here *re-train* means inheriting models trained with data from one survey as the network initialization, and then optimize on data from another survey. For example, I trained a network from scratch with NVSS images and then re-trained the model with FIRST images afterward, and vice versa. The transfer learning methods adopted here do not change the network architecture, the early stopping criteria (10 epochs), or the training sample size and hyperparameters (e.g. batch size and primary learning rate). These methods only specify which layers to freeze, and whether to trigger the training using pre-trained weights.

The purpose of Method 0 is to examine whether transfer learning can provide a better starting point and form a better-trained model. Methods A and B are intended to examine if directly inheriting features from pre-trained convolutional layers can produce comparably good results and reduce training time.

Figure 3.6 shows average learning curves for each of the three methods and includes the 'Xavier' models for comparison. In general, transfer learning constrains the standard deviation compared to direct training by at least a factor of 2. Such an effect is seen regardless of the transfer learning method applied or survey data used. In addition, transfer learning accelerated the convergence of the model in each case. Finally, the application of transfer learning provided a higher starting validation accuracy after the first epoch of training. Among the three methods, Method 0 seems to perform best. By applying Method 0, models share highest starting points for training on both NVSS (98.7%) and FIRST (99.8%) after only one epoch of training. Though the accuracy gap between the 'Xavier' models and the Method 0 models gradually decreases, the models using Method 0 in fact provide better validation accuracy with final values above 99.9%. In the case of independent training stability, the 'Xavier' models have a validation accuracy standard deviation after the final epoch on order of 10^{-3}, whereas the value for models using Method 0 ranges from 10^{-5} to 10^{-4}. The time spent on training for models using Method 0 is slightly longer than training from scratch: randomly initialized models need 33 ms to train on an image, while the time required for Method 0 is about $\sim$34 ms. Generally speaking, each training run takes $\sim$124 minutes on our test system.

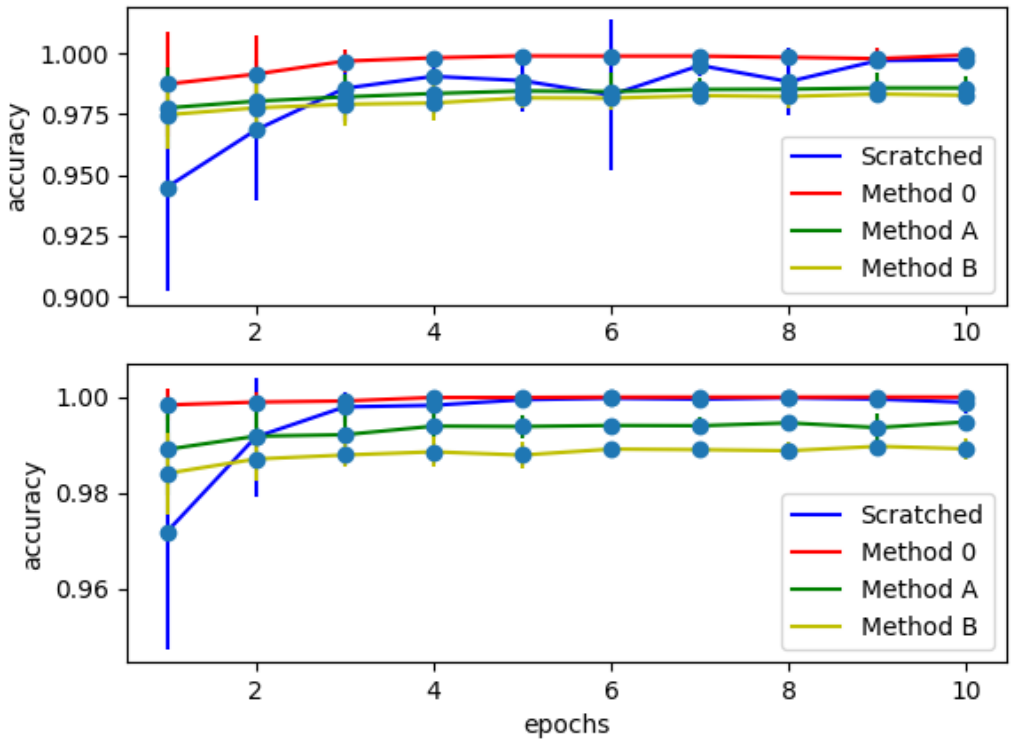


FIGURE 3.6: Upper: Model average validation accuracy curves with corresponding error bars trained with inherit NVSS images using variant methods. Blue, red, green and yellow curves represent models trained from scratch, using Methods 0, A, and B, respectively. Lower: The same curves trained with inherited weights trained on FIRST images.

Comparatively, Methods A and B may require more training over a larger number of epochs as their training accuracies grow very slowly. Models using Method A or B have a validation accuracy which grows from 0.5% to 0.8% at each epoch, while the growth for 'Xavier' models is $> 2.8\%$. This is understandable given that the convolutional layers are frozen. However, freezing these layers reduces the time cost for model training. Both Methods A and B require only 7 ms to train each input image, $\sim$21% of the time cost using Method 0. Comparing the two methods, Method A performs better that Method B due to the difference in its weight inheritance. Further analysis of classification accuracy using the three methods will be given in Sec 5.1.

3.4 Results

3.4.1 Classification Accuracy

Besides validation sets, the prediction-truth comparison is another factor when evaluating a classifier. Accuracy is often considered as the primary parameter for evaluation; however, whilst this metric can provide an overview of model performance, this can be misleading when test samples have a significant class imbalanced.

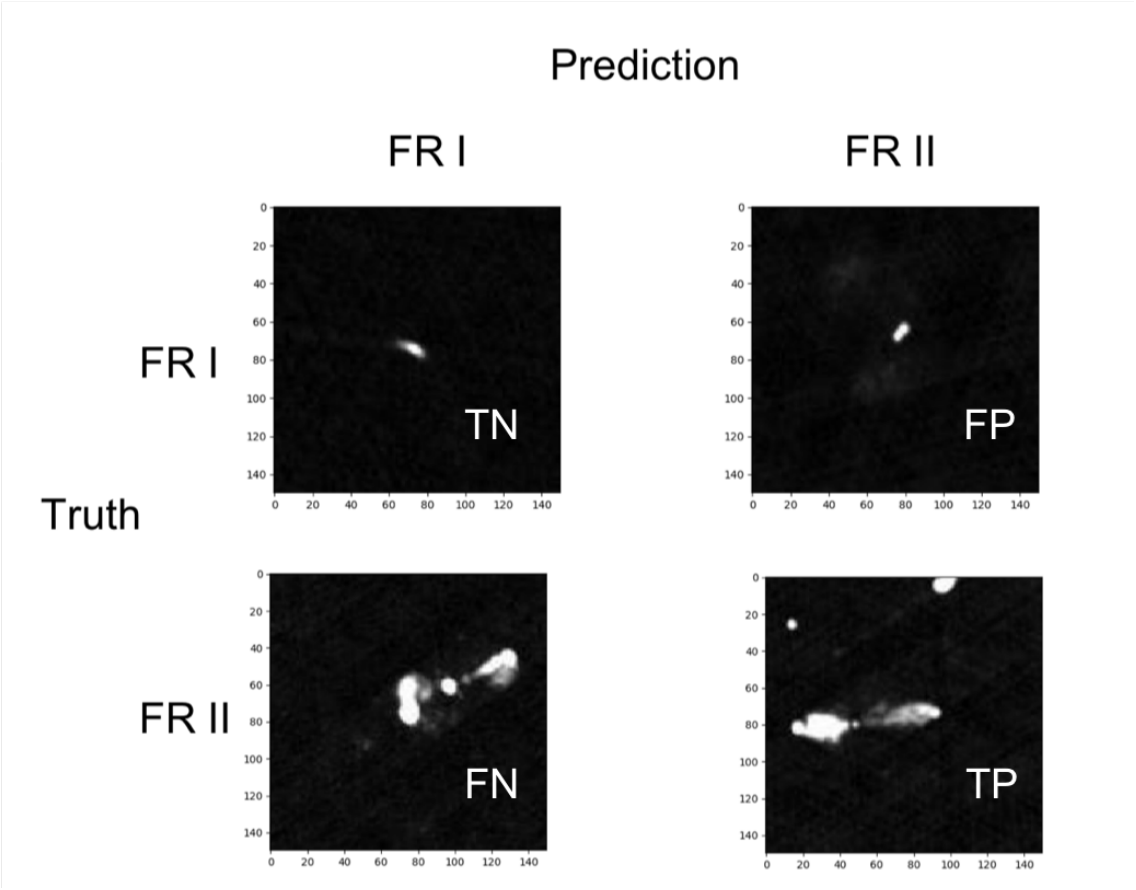


FIGURE 3.7: An example of a confusion matrix. In the context of binary classification, FR I and FR II represent false and true classes. All pre-processed FIRST images in the matrix came from the test set.

A widely used tool for evaluating model performance on a class-wise basis is the confusion matrix (Stehman, 1997), a table to visualize model performance. Figure 3.7 gives an example of a confusion matrix adopted for this work. This matrix has four quadrants: True-Positive (TP), False-Negative (FN), True-Negative (TN), and False-Positive (FP). Assuming FR II morphology as "true," the matrix counts the FR II test samples towards TP if model prediction matches its class label identified by human.

The four-quadrants can be used to further assess the predictive ability of a two class model as the basis for deriving three additional metrics: Recall (R), Precision (P), and F_1 score (Powers, 2008). R is the ratio of TP and TP + FP, while P is TP /(TP + FN). In the context of classifying FR II morphology, a higher R and P score refers to greater sensitivity and class prediction accuracy, respectively. The F_1 score,

$$\mathrm{F}_1 = 2\,\frac{\mathrm{P}\times\mathrm{R}}{\mathrm{P}+\mathrm{R}}$$

can be seen as the weighted average of R and P, which provides a general assessment when identifying samples of a class. These four metrics enable us to evaluate the models trained in this work. Each model was tested with identical NVSS and FIRST image sets.

In addition to the numerical value for each of these four metrics, the Receiver Operating Characteristic (ROC; e.g. Figure 3.8) curve is also used to represent model performance. The ROC curve is mainly a useful tool when making binary classification, although multi-class variants do exist. It provides a visualisation of the false-positive rate versus the true-positive rate for a number of candidate thresholds from 0 to 1. The true-positive rate is equal to the recall when the class being considered is the 'real' class, while Equation 3.4.1 defines the false-positive rate:

$$\text{False positive rate} = \frac{\texttt{fp}}{\texttt{fp}+\texttt{tn}}$$

The ROC curve can be seen as a trade-off between the two variables, and the area under the curve (AUC) value is often used when comparing models. In such a comparison, testing on the same samples, the model with a higher AUC is considered to perform best when distinguishing classes.

3.4.2 Randomly initialized models

Depending on the choice of datasets used in training and testing, classifier performance can vary under evaluation. Figure 3.8 shows the ROC curves for different models trained from scratch. Models trained with NVSS samples show similar behavior when tested on either NVSS or FIRST samples. The average AUC for testing the two sample sets is 0.80 $\pm$ 0.01 and 0.78 $\pm$ 0.02, respectively.

In comparison, models trained using FIRST samples show an asymmetric performance. Such models work well when classifying unlabelled (test) FIRST images, but they behave randomly when tested on NVSS images. The AUC for these models reached 0.94 $\pm$ 0.01 for the FIRST samples, while the metric for NVSS was 0.54 $\pm$ 0.05.

For other metrics the situation is similar. Table 3.3 summarizes the metrics described above for these models. It can be seen that models with higher AUC scores also share higher classification accuracy. Models trained using FIRST images perform best when making predictions on FIRST test images. These models generally achieve 89.1% $\pm$ 1.4% accuracy. When models are trained and tested on NVSS images, however, test accuracy drops to 73.0% $\pm$ 1.1%. Such a change might be attributed to the differences between the two surveys: sample sources in FIRST are well resolved and extended in most cases, sources in NVSS, however, are sometimes only slightly resolved and are small in the image (Figure 3.3).

When models were trained on FIRST images and tested on NVSS images, however, I saw strong FR I class preference in these models. Neither model recall nor precision when classifying NVSS FR II images is higher than 0.5. Such bias also exists when models are trained with NVSS images. All randomly initialized models perform better when classifying NVSS FR I samples. Given that the training data is well balanced, with a 0.45% higher number of FR I samples than FR IIs, it is unlikely that this is a consequence of class imbalance alone.

Since the model trained by Aniyan & Thorat (2017) used similar input data samples, I naively compare the results of our randomly initialized models trained on FIRST images

NVSS trained	**NVSS test result**		**FIRST test result**	
	FR I	**FR II**	**FR I**	**FR II**
Recall	0.67±0.01	0.87±0.04	0.74±0.06	0.70±0.06
Precision	0.92±0.02	0.54±0.03	0.67±0.04	0.77±0.07
F1 Score	0.77±0.02	0.67±0.05	0.70±0.08	0.73±0.10
Accuracy(%)	73.0±1.1		71.9±2.8	
FIRST trained	**NVSS test result**		**FIRST test result**	
	FR I	**FR II**	**FR I**	**FR II**
Recall	0.49±0.01	0.40±0.17	0.85±0.02	0.94±0.04
Precision	0.92±0.02	0.05±0.02	0.95±0.02	0.83±0.04
F1 Score	0.64±0.02	0.09±0.06	0.90±0.03	0.88±0.06
Accuracy(%)	48.5±1.2		89.1±1.4	

TABLE 3.3: A summary of model performance for randomly initialized models trained for 10 epochs. Testing for models trained on one survey images adopted the test image set from the same survey.

to that work. Our precision when classifying FR I objects is $\sim$ 95%, similar to theirs. For FR II classification, our models achieved 83% precision, compared to 75% from their models. The average F1 score in our work is 91%, 5% higher than Aniyan & Thorat (2017); however, the recall of their models for classifying FR Is on the other hand is 6% higher than ours. The difference between these two works may be explained in several ways.

Firstly, the fusion model they proposed was a three-class classification model and a number of bent-tailed radio galaxies were mistakenly identified as FR II sources (Aniyan & Thorat, 2017). Secondly, there is a discrepancy in the number of input data samples and their distribution. When training their models, Aniyan & Thorat (2017) imported 36,000 FR Is, 32,688 FR IIs, and 25,488 Bent-Tailed 'sources', whereas our complete data sample contains 39,796 samples. This might contribute to higher recall. Finally, although I imported FR Is and FR IIs from the same catalogues, the definition of FR Is between the two differs slightly: some FR Is I imported were considered as bent-tailed sources (Aniyan & Thorat, 2017). This could potentially cause a difference in model performance.

3.4.3 Transfer learning models

Although a naive analysis of the transfer learning models presented in this work might initially indicate the advantages of applying these methods, it is important to consider that these models may show varying performance characteristics when applied to new or different unseen datasets. Table 3.4 gives a summary of the test accuracy for models trained with or without transfer learning methods, when applied to a dataset different from that used for original training. Model test accuracy represents the general performance of a classifier.

The application of Method 0 boosted model classification ability when predicting NVSS images and gave the best test performance among the three methods. The boosting effect holds even if I decrease the number of training samples. I naively tested the

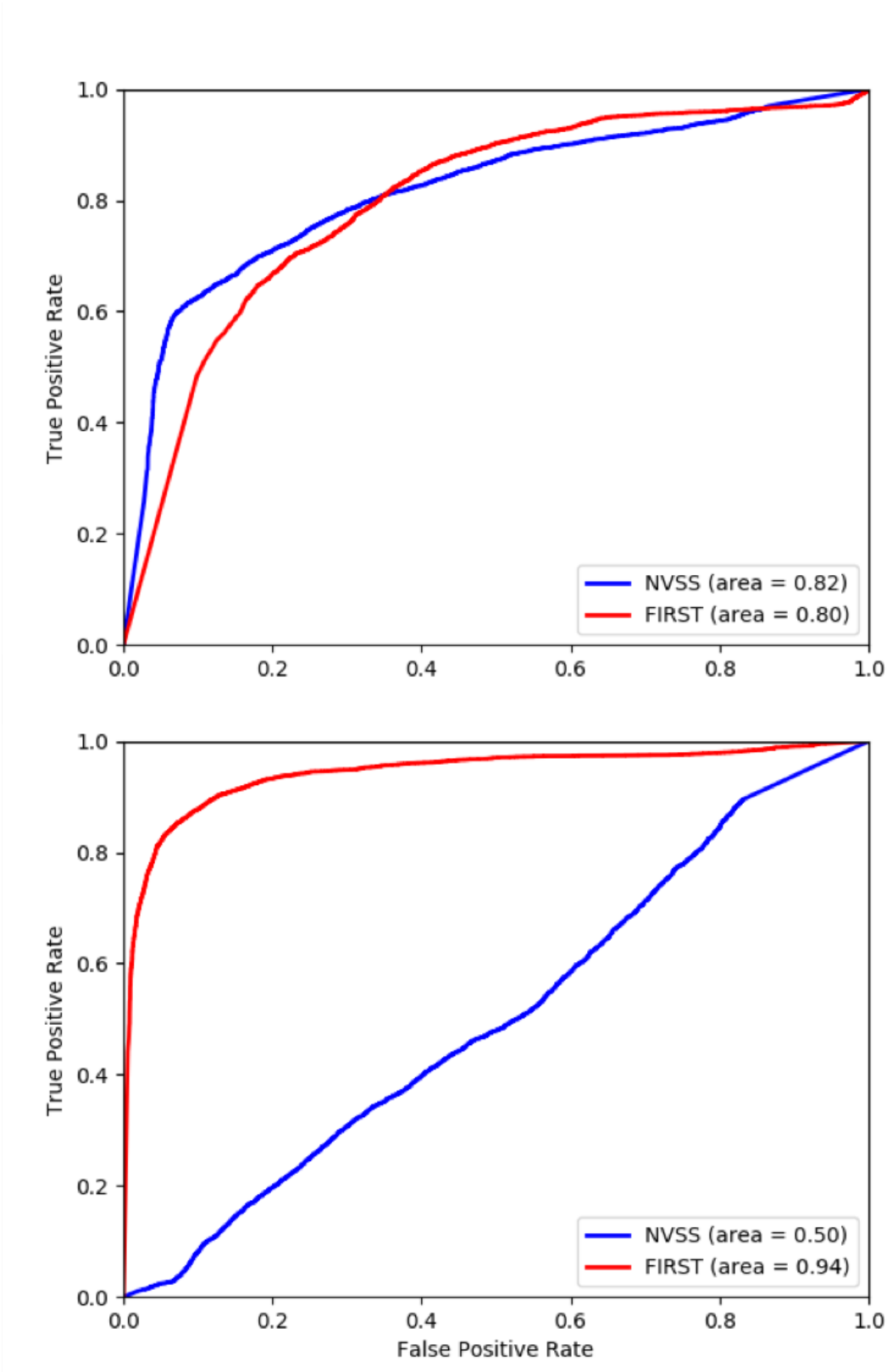


FIGURE 3.8: ROC curve for 'Xavier' models. The colors in the diagram represent the survey of the test images used to derive the curve. Blue refers to NVSS images, while red represents FIRST images. Upper: ROC curves for 'Xavier' models trained on NVSS images for 10 epochs. Lower: ROC curves for 'Xavier' models trained on FIRST images. When deriving the curves, the FR I class is assumed to be "true", while the FR II class is considered to be "false".

NVSS trained	**Xavier**	**0**	**A**	**B**
NVSS test(%)	73.0±1.1	**73.0±0.7**	69.8±1.0	71.6±0.8
FIRST test(%)	71.9±2.8	78.4±2.0	**81.1±1.0**	78.3±1.0
FIRST trained	**Xavier**	**0**	**A**	**B**
NVSS test(%)	48.5±1.2	**50.3±0.7**	46.9±0.5	46.2±0.6
FIRST test(%)	**89.1±1.4**	87.4±1.4	84.6±0.6	83.8±1.0

TABLE 3.4: A summary of averaging model accuracy. Accuracy in the table are represented in percentage. 'trained' refers to the survey data finally trained on each model. Bold implies that the method horizontally gave the best accuracy.

possibility in reducing training samples of transfer learning: only rotate all NVSS sample images by 36, 72, 108, ... degree. This generated a smaller dataset of 6590 images. This smaller dataset contains 3690 images in the training set, which equals to 16.5% of training samples in our primary NVSS dataset. I applied Method 0 using this dataset, and tested these models with our primary NVSS and FIRST datasets. It turns out that these models give 78.2% $\pm$ 2.4 of accuracy when testing on FIRST images. When testing on NVSS models, these model reached accuracy of 73.5% $\pm$ 0.8.

When classifying FIRST images, however, models which inherited pre-trained weights from FIRST images and applied Method A performed best. The same result is not true in the case where the order of the survey data used for the inherited-retraining sequence was switched from FIRST to NVSS.

The AUC values also provide further detail. Models which used Method 0 showed comparatively stronger expressive ability than randomly initialized ones. Adopting Methods A and B produced a similar effect when inheriting weights from models pre-trained on FIRST images. Such a phenomenon, however, lost its efficacy if inheriting weights from models initially trained on NVSS images. This can perhaps be explained by the difference between the two surveys. Images of many sources seen in the FIRST survey possess richer structural information than provided by NVSS. This is an important factor when considering the application of pre-trained models from existing surveys to new data from next-generation telescopes such as ASKAP, MeerKAT and the SKA.

In addition to accuracy and AUC values, I also evaluated these transfer learning models further by measuring their recall, precision, and F1 score for each class. Figs 3.9 & 3.10 show how transfer learning models behave when classifying either FR Is or FR IIs on NVSS and FIRST images. I note that models consistently identified most test samples as FR Is if they were re-trained with FIRST inputs and tested using an NVSS test dataset. When models are re-trained with NVSS images I found that introducing transfer learning improved both FR I and FR II classification for NVSS images. The identification of FR II objects typically reached 88% precision using Method A. This is $\sim$ 5% higher than models trained with FIRST images directly. In the context of FR I classification using FIRST samples, the highest achieved precision value was 95%. Such precision could either be

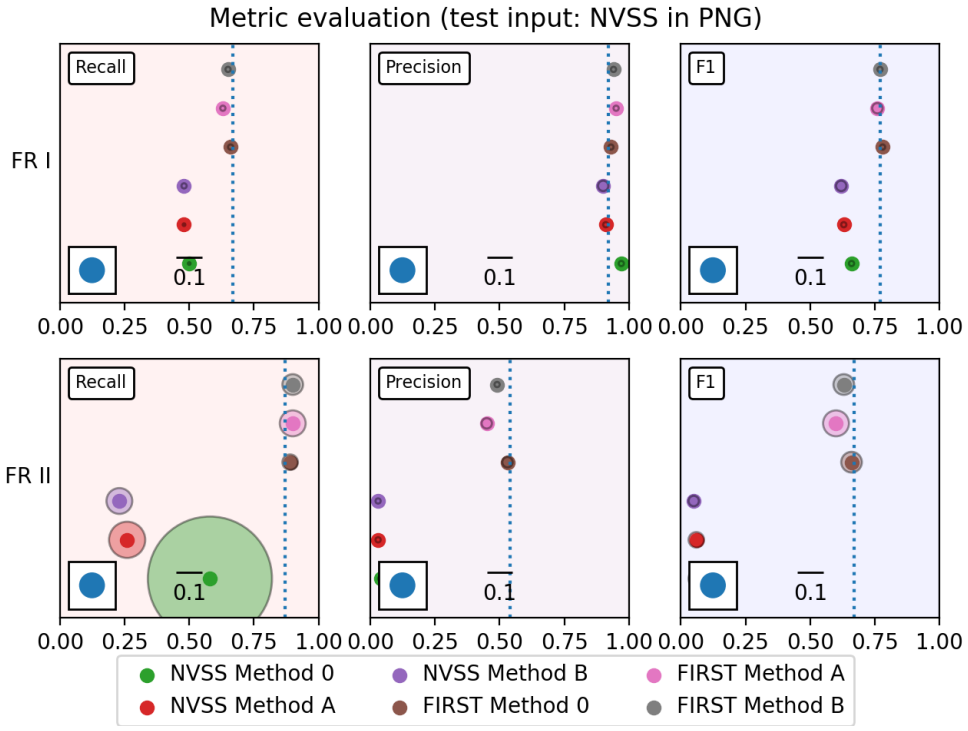


FIGURE 3.9: A summary of metric evaluation for models applied transfer learning and tested on NVSS images. 'NVSS' or 'FIRST' shown in the legend box implies that, when applying transfer learning, the pre-trained model weights were trained on the named survey. In the diagram, radius of the circles accounts for the standard deviations of their respective metrics. Dashed vertical lines refer to average metrics for the Xavier models trained and tested on NVSS images.

achieved through direct training or by using Method 0 when training with FIRST images.

These results imply that choice of method is a trade-off. Applying Method 0 would strengthen model stability and boost the performance of a model if one wished to apply the model to both NVSS and FIRST images. Method A can boost prediction precision when identifying FR IIs on FIRST images. If not training FIRST images from scratch, applying Method A can make the most precise prediction on FIRST images. Method B can be seen as an alternative option in the case where computational power is constrained and one wants a quickly trained model which makes a reasonably good prediction.

3.4.4 Influence of input image format

The input images used for model training and testing described in Section 3.4.2 and Section 3.4.3 are processed in PNG image format. When saving images, such a format converts the value of each pixel to an integer in a lossless fashion.

In addition to PNG format, many classifiers also accept images in JPEG format. The advantage of using JPEG images is that they can require smaller storage volume and

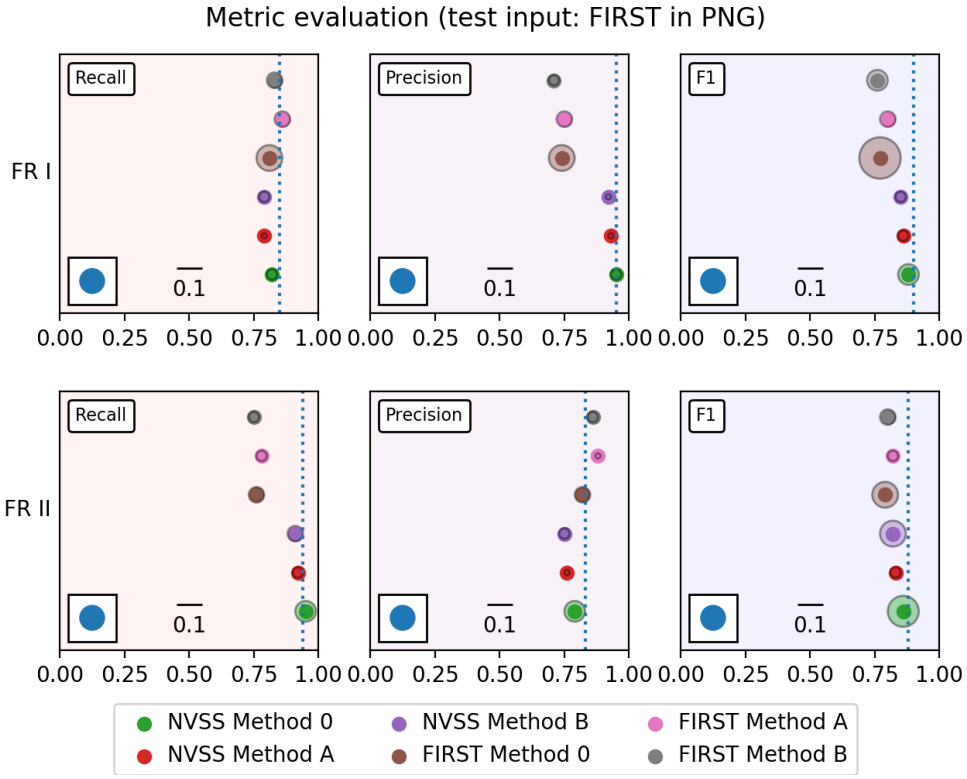


FIGURE 3.10: A summary of metric evaluation for models applied transfer learning and tested on FIRST images. Models evaluated in the diagram are the same as Figure 3.9. The meanings of symbols and texts in the diagram are consistent to Figure 3.9 as well. Dashed vertical lines refer to average metrics for the Xavier models trained and tested on FIRST images.

NVSS trained	**Xavier**	**0**	**A**	**B**
NVSS test	0.80±0.01	**0.81±0.01**	0.80±0.01	0.81±0.00
FIRST test	0.78±0.02	0.86±0.01	**0.88±0.01**	0.83±0.01
FIRST trained	**Xavier**	**0**	**A**	**B**
NVSS test	0.54±0.05	**0.59±0.02**	0.57±0.01	0.53±0.01
FIRST test	0.94±0.01	**0.94±0.00**	0.93±0.00	0.92±0.00

TABLE 3.5: A summary of averaging model AUC. 'trained' refers to the survey data finally trained on each model. Bold implies that the method horizontally gave the highest AUC.

NVSS trained	**Xavier**	**0**	**A**	**B**
NVSS test(%)	82.1±4.4	**83.9±1.2**	81.8±0.6	82.6±0.6
FIRST test(%)	65.5±7.4	**73.7±2.0**	65.9±1.0	66.0±1.0
FIRST trained	**Xavier**	**0**	**A**	**B**
NVSS test(%)	**57.8±1.9**	57.7±1.2	56.6±0.5	57.2±1.7
FIRST test(%)	**90.1±1.0**	89.7±0.8	87.6±0.7	87.2±0.9

TABLE 3.6: A summary of averaging model accuracy. Model inputs considered in the diagram are in JPEG image format. Accuracy in the table are represented in percentage. 'trained' refers to the survey data finally trained on each model. Bold implies that the method horizontally gave the best accuracy.

have enhanced smoothness. However, when converting image arrays to JPEG format, the images are compressed and there is information loss.

The influence of input image format on model performance has not been addressed in the context of radio galaxy classification. However, the issue of archival data storage for the next generation of radio telescopes may have implications for training data availability. In order to investigate the effect of image format I repeated the image pre-processing, data augmentation, model training, and transfer learning processes described above using images input in JPEG format in order to compare the resulting model outcomes with those using PNG inputs.

Table 3.6 summarizes model performances using image inputs in JPEG format. Models using JPEG inputs show stronger identification ability. Comparing with Table 3.4, classifiers primarily trained with NVSS images showed a 9% accuracy improvement when classifying NVSS test sets. For those models trained with FIRST images, however, classification accuracy is boosted for both NVSS and FIRST test datasets.

When considering Figures 3.11 & 3.12, it can be seen that the F1 score of `Xavier` models when classifying NVSS FR Is and FR IIs shares a common improvement. Typically, these models have their FR II identification ability strengthened significantly. The precision of FR II classification on NVSS test images reached 79%, while the number when using PNG input was only 54%. Nevertheless, when considering FIRST images, the models showed a balanced but relatively smaller recall, precision, and F1 score.

The `Xavier` models trained with FIRST images also showed general improvement when identifying NVSS images. Though the issue of FR I preference still exists, recall of FR IIs classification increased by 39% compared to that using PNG inputs. This implies that by using JPEG images, the classifier achieved a higher sensitivity for identifying FR IIs.

The JPEG-based results also echo the transfer learning outcomes seen using PNG format. No matter which method was applied, models inheriting weights trained on FIRST images and then re-trained on NVSS images make a more accurate prediction. Typically, by applying Method 0, models work optimally for classifying both NVSS and FIRST images. When transfer learning models inherited weights from models trained on FIRST

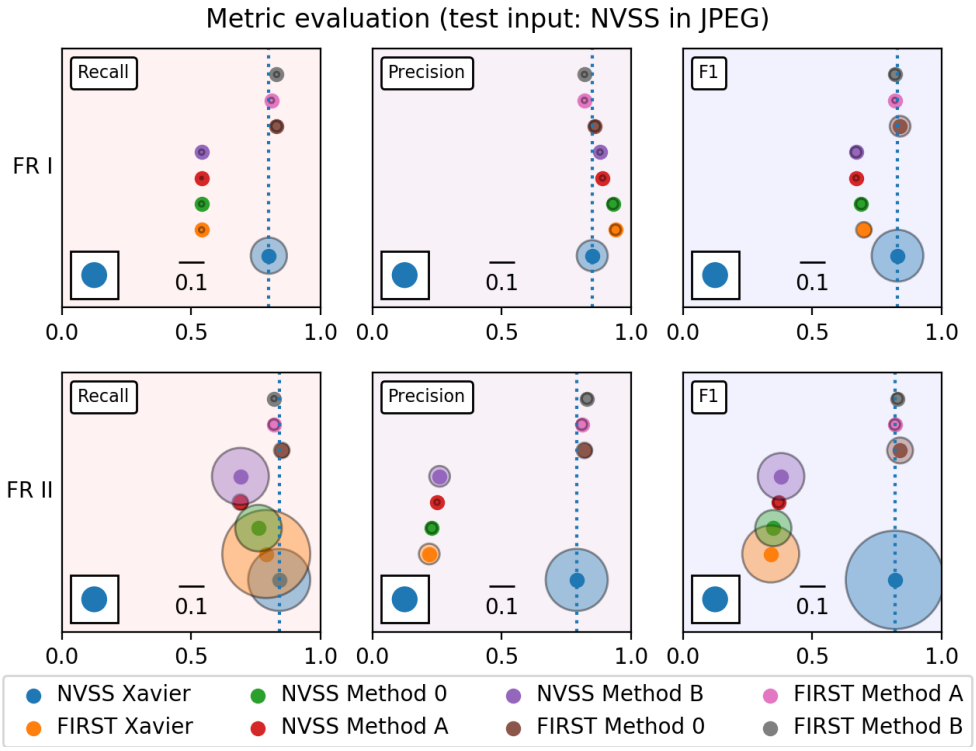


FIGURE 3.11: A summary of metric evaluation for models applied transfer learning and tested on NVSS images in JPEG input format. For transfer learning models, 'NVSS' or 'FIRST' shown in the legend box implies that, the pre-trained model weights were trained on the named survey. For 'Xavier' models, however, survey name refer to the survey data used in model training. In the diagram, the radius of the circles accounts for the standard deviations of their respective metrics. Dashed vertical lines, on the other hand, represent the average metrics for Xavier models trained and tested on NVSS images.

images, their performance is similar to that using PNG inputs.

In spite of the similarities, I note that there are two other phenomena worth mentioning. The first is that the difference between randomly initialized models and those using Method 0 are reduced when using JPEG inputs. The accuracy difference between the two is less than 0.5%, while the difference is larger than 1.5% using PNG formatted inputs. The second phenomenon is caused by applying Method A. The application of Method A no longer makes the best FIRST prediction if re-trained on NVSS images in JPEG format. If one adopted image input in JPEG when learning and testing, Methods 0 and B would become better options.

Why changing image format leads to overall model performance enhancement is not immediately obvious. To explain it, I consider the different input images from the perspective of their information content. I do this by evaluating the Shannon entropy of input images in FITS, PNG and JPEG formats. It it noted that all images have experienced sigma-clipping.

Shannon entropy refers to the averaged self-information content of a dataset (Shannon & Weaver, 1949). Self-information can be defined as the probability that a stochastic

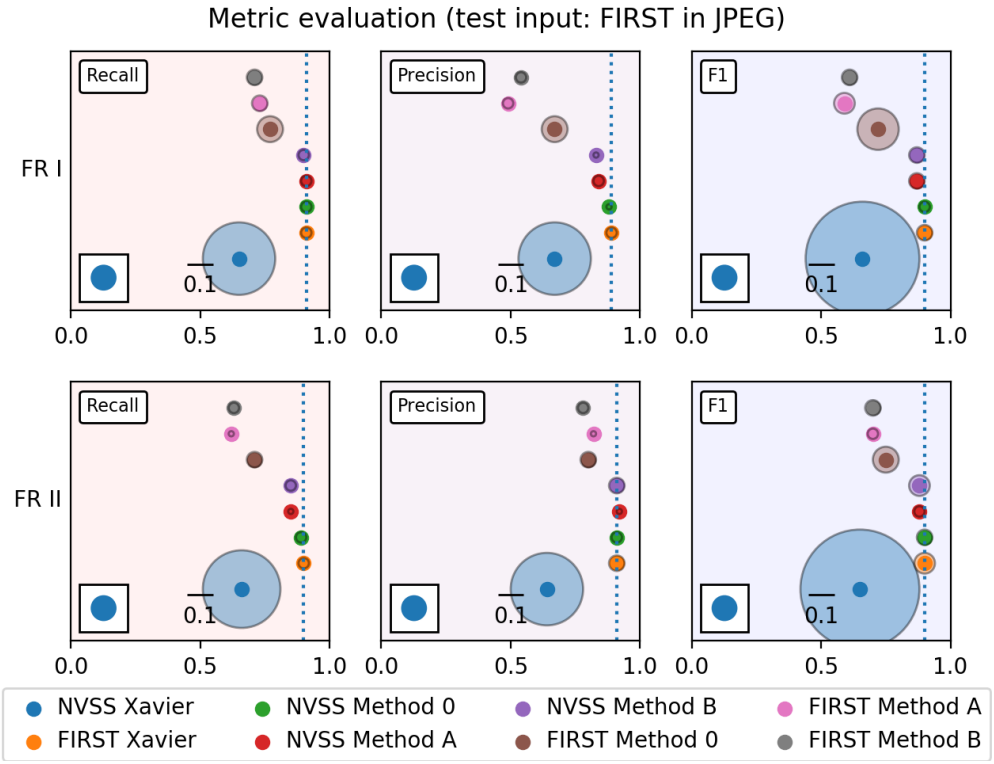


FIGURE 3.12: A summary of metric evaluation for models applied transfer learning and tested on FIRST images in JPEG input format. Models evaluated in the diagram are the same as Figure 3.11. The meanings of symbols and texts in the diagram are consistent to Figure 3.11. Dashed vertical lines, on the other hand, represent the average metrics for 'Xavier' models trained and tested on FIRST images.

source of noise has produced the information in the dataset. Equation 3.4.4 gives the mathematical definition of Shannon entropy, S,

$$S = -\sum p_k \log p_k$$

where p_k represents the normalized pixel values considered as probabilities. For this work, I adopt 2 as the logarithmic base when measuring Shannon entropy.

By definition, inputs with lower Shannon entropy have smaller variation . Also, since the image inputs in this work are normalized to the same pixel range $(0 - 255)$, an image with high Shannon entropy should have a weakly concentrated pixel value distribution. In other words, a model would find it easier to learn image pixel value gradients if the same image had higher Shannon entropy.

I compared mean Shannon entropy between inputs in different data formats from different surveys. Table 3.7 provides a summary of these entropy measurements. In general, NVSS sample images have higher Shannon entropy than FIRST sample images. When I take Table 3.4 and Table 3.6 into account, I find that most models re-trained with FIRST images tended to have a smaller standard deviation in accuracy.

When converting input images from the CoNFIG catalogue from FITS format to JPEG or PNG, I found that their Shannon entropy consistently dropped. Images in PNG format have the lowest mean Shannon entropy of all three formats. In the context of classification, models seem to make a more accurate predictions if the input data shares higher mean Shannon entropy. Since inputs in FITS format have the highest Shannon entropy, it is recommended that future networks use FITS inputs for both training and testing machine learning models.

Regardless of survey or catalogue, FR II inputs experiencing format conversion show higher fractional loss of entropy than FR Is. However, I did not see a relationship between this loss and model performance. When I train and test model using data from the same survey, Recall, Precision, and F1 score differences between the two classes are less than 10%. Such differences become more apparent when testing on a different survey, but the self-information imbalance between the two classes is not sufficient to explain the difference. When applying transfer learning, models continued to give comparative or more accurate FR II and FR I predictions on FIRST and NVSS images, respectively.

Overall, how image formats affects model performance still requires further investigation. Whether Shannon entropy can be seen as an evaluation factor also needs examination in the future.

3.4.5 The application of transfer learning to future radio surveys

Traditionally, radio galaxy classification has been done by visual inspection, sometimes facilitated by measurement of host-hotspot relative positions. Such a method was practical due to the modest sample size of archival catalogues. However, soon next-generation radio catalogues such as that from the EMU survey (Norris et al., 2011), will discover millions of radio sources waiting for visual inspection.

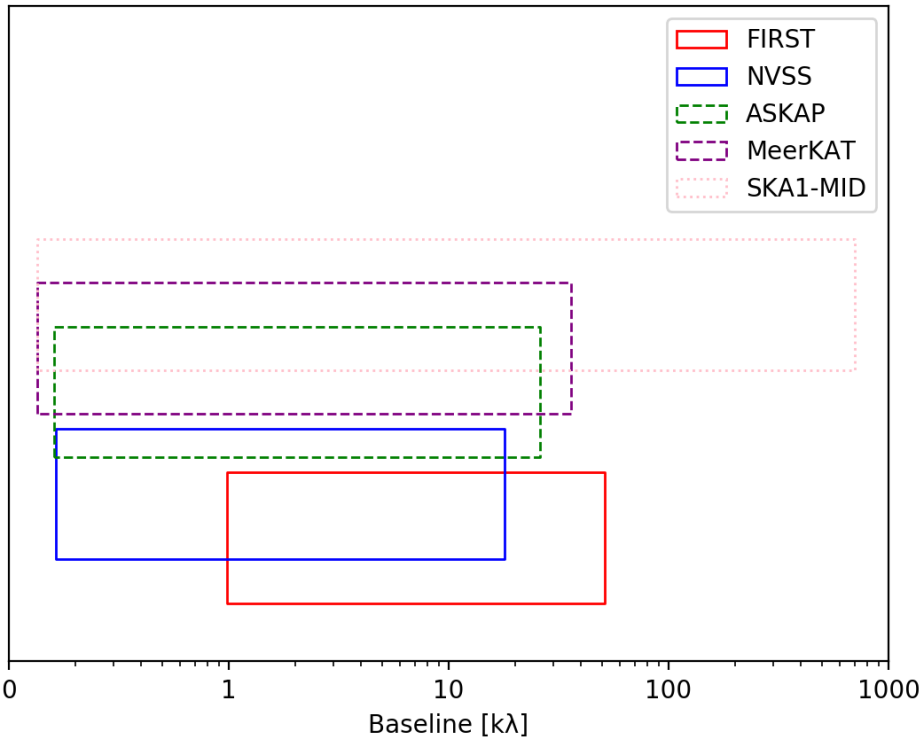


FIGURE 3.13: A summary of spatial scales for several radio telescopes/surveys in units of kilo-lambda (kλ). Solid: finished radio surveys. Dashed: radio telescopes (almost) finish construction. Dotted: telescope would be built in the future. Spatial scales shown in the diagram are converted from telescope baselines in units of `km`. The frequency adopted when doing the conversion is 1.4 GHz for FIRST, NVSS, MeerKAT and SKA1-MID. I adopted 1.3 GHz for ASKAP specifically for its EMU survey (Norris et al., 2011). FIRST was observed using the VLA B-configuration of the VLA (Becker et al., 1995), while NVSS adopted the more compact D and DnC configurations of the same array (Condon et al., 1998). ASKAP have minimum and maximum baseline of 37 m and 6 km, respectively (Johnston et al., 2008; Serra et al., 2015). Baselines of MeerKAT ranges from 29 m to 7 km (Jonas & MeerKAT Team, 2016). Finally, SKA1-MID is expected to have 150 km maximum baseline. The shortest baseline of SKA1-MID here is the same as MeerKAT, as MeerKAT will finally become a part of SKA1-MID core (Serra et al., 2015).

NVSS inputs	**FITS**	**PNG**	**JPEG**
CoNFIG FR I	0.32±0.12	0.18±0.11	0.22±0.14
CoNFIG FR II	0.28±0.03	0.15±0.7	0.19±0.1
FRICAT	0.28±0.03	0.20±0.06	0.28±0.11
FIRST inputs	**FITS**	**PNG**	**JPEG**
CoNFIG FR I	0.25±0.16	0.15±0.09	0.19±0.11
CoNFIG FR II	0.14±0.09	0.06±0.05	0.07±0.07
FRICAT	0.09±0.05	0.07±0.03	0.19±0.09

TABLE 3.7: A summary of Shannon entropy measurement for image inputs in different formats. Shannon entropy for inputs in FITS, JPEG, and PNG format have all experienced image pre-processing.

In order to overcome the difficulties of classifying these sources by eye, recent studies have focused on developing machine-learning based automated methods to classify radio source morphologies based on specific radio surveys. In this chapter, I have introduced the next step in the use of these methods and explored the possibility to boost model performance by applying transfer learning.

Our approaches achieved over 90.1% and 83.9% in terms of classification accuracy when testing on FIRST and NVSS images, comparable with other recent state-of-the-art results. Depending on the transfer learning method used, I have demonstrated that transfer learning models can result in even higher model accuracies or save training time by up to 79%.

A key result from this work is that inheriting model weights pre-trained on higher resolution survey data, e.g. FIRST, can boost model performance when re-training with lower resolution survey images, e.g. NVSS. However, I found that the reverse situation, whereby weights inherited from models trained on lower resolution data are re-trained on higher resolution data, is detrimental to model performance. Such model performance perhaps could be explained according to the number of sample morphological features available to be captured by a model: High resolution data usually contains more morphological information of a sample object comparing with those data with lower resolution, and hence allow a model to extract more specific sample features from them. It is then unsurprising that a model learned from these features can boost model performance when performing re-training with lower resolution survey data.

This is of particular relevance for future radio surveys, where machine learning weights inherited from models trained on archival data may be used to initiate classifiers for previously unseen data. Figure 3.13 summarizes the baseline ranges of the NVSS and FIRST surveys, along with the ranges of ASKAP, MeerKAT, and SKA1-MID. These three telescopes are capable of making observations at 1.3-1.4 GHz, similar to FIRST and NVSS which were made at 1.4 GHz. It can be seen that there are considerable spatial scale overlap between MeerKAT, ASKAP and the surveys considered in this work.

The higher resolution of the FIRST survey, relative to both MeerKAT and ASKAP

as well as NVSS, suggests the potential for successful transfer learning approaches to machine learning classification of the survey data from these next-generation telescopes. However, an issue should be created carefully before applying transfer learning approaches: survey depth. Most data samples considered in this work have their host galaxy redshift smaller than 1 (Gendre et al., 2010; Capetti et al., 2017a), while next generation surveys are likely to discover AGN to the edge of visible Universe (i.e. Norris et al., 2011, EMU). A classifier able to identify sample morphology at local universe might lose its power when facing objects of higher redshifts. Further investigation should be considered carefully before one applies transfer learning on these surveys.

Finally, even if the survey depth issue has been solved, the significantly improved resolution of SKA1-MID in comparison suggests that further investigations must also be made before the advantages of transfer learning can be used there.

Chapter 4

Identification of New Giant Radio Galaxies

The work in this chapter is published in Tang et al., 2020, *Monthly Notices of the Royal Astronomical Society*, Volume 499, Issue 1, pp.68-76.

In this chapter, I describe the discovery of five new giant radio galaxies selected from the Radio Galaxy Zoo Data Release 1 (RGZ DR1; Wong et al., in prep.). RGZ DR1 is a manually cross-matched radio galaxy catalogue, using the efforts of more than 12,000 citizen scientist volunteers (Wong et al., in prep.). Unlike previous GRG identification studies, this work uses a process compatible with the constraints imposed by current deep learning algorithms. In Section 4.1 I describe the initial source selection process; in Section 4.2 I describe the validation process for identifying GRGs; in Section 4.3 I draw comparisons with other GRG identification studies, including their comparative selection effects and resulting impact on deep learning algorithms. I also discuss the characteristics and environments of these newly identified GRGs in a wider context; and in Section 4.5 I summarise the conclusions from this chapter.

This work has assumed a ΛCDM cosmology with Ω_m = 0.31 and a Hubble constant of $H_0 = 67.8$ km s^{-1} Mpc^{-1} (Planck Collaboration et al., 2016). The AllWISE magnitudes adopted in the work follow the Vega magnitude system (Wright et al., 2010). Finally, I adopt the radio spectral index convention to be $S \propto \nu^{\alpha}$ throughout the work, where α is the spectral index.

4.1 Source selection

RGZ DR1 is a catalogue from the first 2.75 years of the radio galaxy zoo project (Wong et al., in prep.). Within the catalogue, 99.2% of classifications used radio data from the FIRST survey, and the remainder used data from the ATLAS survey. Each source classification has a user-weighted consensus fraction (consensus level) > 0.65 (Wong et al., in prep.). The LAS of each source in the RGZ DR1 is estimated by measuring the hypotenuse of a rectangle that encompasses the entire radio source at the lowest radio contour (Banfield

et al., 2015). This method is generally reliable if the radio lobes of a source are correctly identified and the source is not severely bent.

The RGZ DR1 catalogue contains information on individual radio galaxies and their associated radio components, and also a table of cross-matched host galaxies. In this study, I used the catalogue of cross-matched host galaxies as our primary input sample. The original catalogue contains ∼140,000 entries. From this catalogue I removed sources without FIRST data available or without host galaxy spectroscopic redshift data in the Sloan Digital Sky Survey (SDSS DR12; Alam et al., 2015). This reduced the sample to 11 549 entries.

Although the RGZ DR1 catalogue requires entries to have a minimum consensus level of 0.65, I found a number of duplicate entries that had (i) the positions of two radio sources separated by less than the pixel size of the FIRST survey (1.8 arcsec), or (ii) multiple host galaxies identified within the same extended radio source. I identified 186 of the first instance and 147 of the second instance. I visually inspected all these source pairs. I then eliminated one source from each pair of the first instance if their LAS and host galaxy redshift were identical. In the second instance, I retained the source in each pair which had a position closer to the WISE host galaxy position. This process removed a further 312 objects, which reduced the number of objects in the sample to 11 237 entries.

Figure 4.1 shows the size-luminosity diagram for this sample. The projected linear size and 1.4 GHz radio luminosity for each object were calculated using the catalogued RGZ DR1 source LAS, integrated flux density, and the host galaxy redshift. Within the sample there are 17 objects which have a projected physical size greater than 700 kpc. The FIRST images for each of these entries were visually inspected, and one additional repeated object was identified and removed. I then crossed-matched the remaining 16 objects with the GRG catalogues of Kuźmicz et al. (2018b), Dabhade et al. (2020a) and Kozieł-Wierzbowska et al. (2020b) and found that Kuźmicz et al. (2018b) had previously recorded three of the objects: J0929+4146, J1511+0751, and J1521+5105. The remaining 13 candidate objects were not found to match any previously known GRG. I also cross-matched the recent Proctor (2016) inspired GRG candidate catalogue from Dabhade et al. (2020c), and found no overlap. The remaining 13 candidate objects were not found to match any previously known GRG.

4.2 Giant Radio Galaxy Identifications

For the 13 candidate GRGs, I refined their identifications and measurements using manual inspection of the data. This inspection was used to clean the dataset in three steps:

1. I examined the relationship between the radio structure and the assigned infrared host galaxy of each entry using a *WISE* 3.4 μm image centred on the estimated central radio emission position of the radio galaxy. This process removed 3 objects where no clear relationship between the radio lobes and the host was seen.

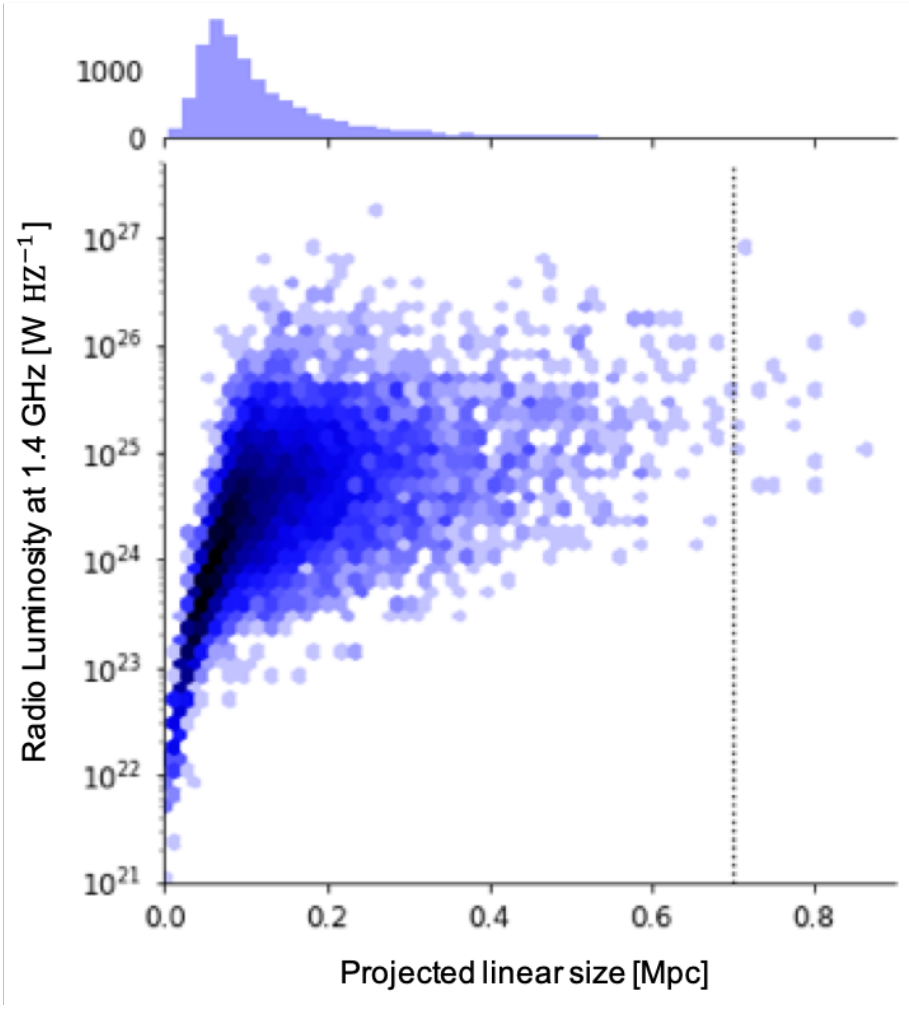


FIGURE 4.1: Upper: The projected linear size histogram of the adopted 11 237 RGZ DR1 candidates. Lower: The size-radio luminosity diagram of the same candidates. The color of each hexagon in the diagram represents the source number density with corresponding size and radio luminosity at 1.4 GHz. The dashed line refers to 700 kpc of linear size.

2. I re-calculated the LAS of each radio galaxy using the HEALPix Ximview software (Górski et al., 2005) and the FIRST images, and compared these values with the LAS recorded in the RGZ DR1 catalogue. This process identified three fields containing two radio galaxies that had been misidentified as a single source. In two further fields, I found that the source LAS was overestimated due to confusion with neighbouring sources and that this had also caused the host galaxy to be misidentified. These five objects were removed from the candidate list.

3. For objects with misidentified host galaxies (point (i) above), I made a renewed host galaxy search using the NASA Extragalactic Database (NED) and the SDSS Sky Server (Aguado et al., 2019). The mid-point of the radio emission was chosen to be the search centre in each case. Given that each image had a side of 3 arcmin, I searched within a radius of 1.5 arcmin. In those cases where a host redshift was found in SDSS DR15, I re-measured the projected linear size of each radio source. This check showed that none of the misidentified sources had a projected linear size larger than 700 kpc.

This three-step data cleaning process resulted in a final sample of 5 GRGs. Figure 4.2 shows the images of these sources; Table 4.1, Table 4.2 and Table 4.3 summarize the redshift, LAS, linear size, infrared and radio properties of each object. Redshifts in the tables are extracted from SDSS DR15. Source LASs have been manually re-measured, but are generally consistent (typically 0.5% larger) with those from the original DR1 catalogue.

For the newly identified GRGs, I used visual inspection of the FIRST data to classify each source by morphology and found four of the five objects to have FR II type morphology (Fanaroff & Riley, 1974). The fifth source, J1646+3627, has an ambiguous morphology.

All five sources have comparatively high radio luminosities, with $\log P_{1.4}\,[\mathrm{W/Hz}] > 25.1$, the mean total radio luminosity of FR II objects as determined by Kozieł-Wierzbowska & Stasińska (2011). Since the host galaxies in each case have W1-W2$<$ 0.8 and W2-W3$<$ 3.5, where W1, W2, and W3 are the *WISE* observed source magnitudes at 3.4 ¯m, 4.6 ¯m and 12 ¯m (Cutri & et al., 2013), they are likely to be either elliptical or intermediate disk galaxies (Jarrett et al., 2017).

The five GRG sources are:

J0941+3126 This source is also known as B2 0938+31A, and is centred at J 9h41m01.24s +31°26′32.3′
(Colla et al., 1970, 1972, 1973; Fanti et al., 1974). The source is hosted by SDSS J094103.62+312618.7. The source has integrated flux density of 20 mJy at 15.2 GHz (Waldram et al., 2010), and 7.2±3.3 mJy at 30 GHz (Gawroński et al., 2010). Its host has $W2 - W3 > 2$, redder than is typical for elliptical galaxies and more consistent with the 'intermediate disk galaxy' designation of Jarrett et al. (2017). The host galaxy in this case currently has only photometrically determined redshifts (Alam et al., 2015; Bilicki et al., 2016; Zou et al., 2019), ranged from 0.282 to 0.398. I adopted the lowest one measured by SDSS DR12. I consequently believe this source can be identified as a GRG for certain.

GRG	RGZ ID	RA (J2000.0) [h:m:s]	DEC (J2000.0) [°:′:″]	z	LAS [arcsec]	Size [kpc]
J0941+3126	J094103.6+312618	09:41:03.62	+31:26:18.7	0.282 ± 0.0454^{p}	163	717 ± 88
J1331+2357	J133117.9+235700	13:31:18.01	+23:57:00.4	0.33610 ± 0.00006^{s}	162	803 ± 7
J1402+2442	J140224.3+244226	14:02:24.25	+24:42:24.3	0.337 ± 0.032^{p}	173	810 ± 12
J1421+1016	J142142.6+101626	14:21:42.68	+10:16:26.3	0.37392 ± 0.00003^{s}	144	765 ± 6
J1646+3627	J164642.5+362710	16:46:42.58	+36:27:10.6	0.43425 ± 0.00010^{s}	130	$>754 \pm 1$

TABLE 4.1: A summary of the newly discovered GRGs found in the present work. RGZ ID for each source represents the truncated host galaxy coordinates recorded in the RGZ DR1 catalogue. RA/DEC of source host galaxies are that of the infrared host galaxies shown in the Figure 4.2. The LAS of the sources is measured using **HEALPix Ximview**. For the first four sources, I have assigned errors of 5 arcsec (FWHM) to the LAS of each source, since their leading edges are fairly sharp. In the case of J1646+3627, I have listed the size as a lower limit as the source could be found to extend further given observations with improved sensitivity to larger scale structure. Redshift annotations: p: photometric; s: spectroscopic.

J1331+2557 This source is also known as 7C 1328+2412, and is centred at J 13 31 18.12 +23 57 07.4 (Waldram et al., 1996). Its north-east lobe is also known as TXS 1328+242. The host galaxy of this source is identified as SDSS J133118.01+235700.4. I found the source has been observed at 1.4 GHz by the VLA archival public project AG0635 (Figure 4.3). The observation has angular resolution of 19.7 × 13.7 arcsec, along with source flux density of 172±8 mJy. The observation shows the source has visible radio core emission cross matched with its host galaxy, and its core flux density is 6±3 mJy. Similarly to J0941+3126, the host galaxy of the source has $W2 - W3 > 2$.

J1402+2442 This source is also known as B2 1400+24, and is centred at J 14h02m25.87s +24°41′53.0″ (Colla et al., 1970, 1972, 1973; Fanti et al., 1974). The host of this source is a close pair of galaxies, SDSS J140224.25+244224.3 and SDSS J140224.31+244226.8. The latter has a photometric redshift $z = 0.299 \pm 0.067$ (Alam et al., 2015). I note that although I identify the above galaxy pair as the host for this source, SDSS J140225.03+244218.1 is also in close proximity, see Figure 4.2. This source has a photometric redshift of $z = 0.208 \pm 0.018$ (Alam et al., 2015).

J1421+1016 This source is also known as MRC 1419+104, and is centred at J 14h21m42.03s +10°16′17.3″ (Large et al., 1981, 1991). This source was mentioned by Amirkhanyan et al. (2015), but not previously identified as a GRG due to differences in the estimation of both the LAS and redshift. This source has a host galaxy SDSS J142142.68+101626.2, which is not visible in Figure 4.2 where I show the SDSS-g image, but can be seen clearly in the WISE 3.4 μm data.

J1646+3627 The host galaxy of this source is 2MASX J16464260+3627107. It is the brightest cluster galaxy in the galaxy cluster GMBCG J251.67741+36.45295 (Hao et al., 2010) and has a slightly bent morphology, see Figure 4.2. This morphology is consistent with the findings of Garon et al. (2019) who used 4304 extended radio sources from RGZ to determine that BCGs have higher probabilities than other cluster members to have slightly bent morphologies.

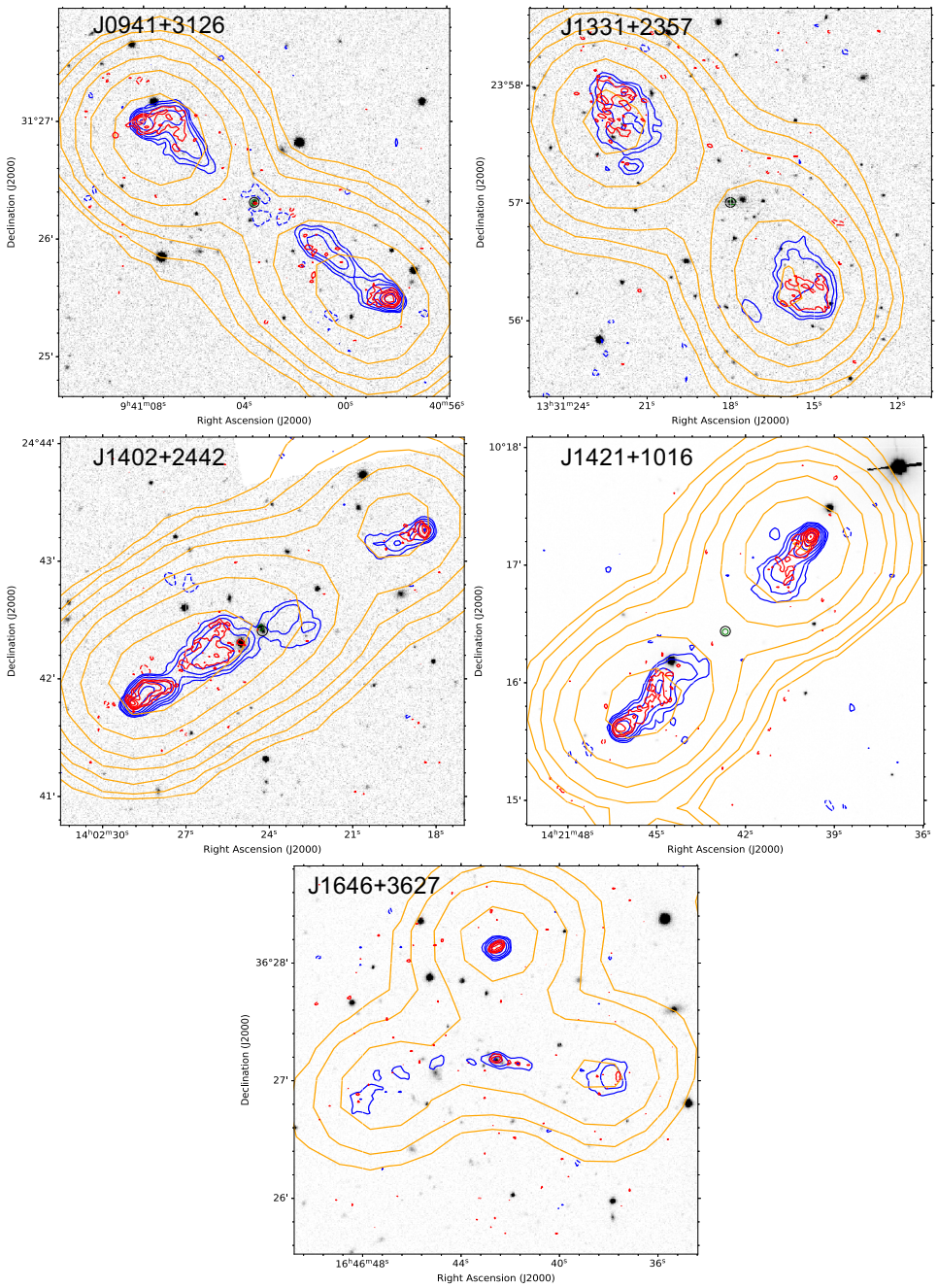


FIGURE 4.2: The new GRGs identified in this work. The figure shows radio-near infrared overlays of these sources, using SDSS i-band images rather than WISE, given their better angular resolution. The orange, blue and red radio contours for each source from the NVSS, FIRST and the Karl G. Jansky Very Large Array Survey (VLASS; Lacy et al., 2020), respectively, are shown on each image from $3\sigma_{\rm rms}$ increasing in steps of 2. The dashed lines are -$3\sigma_{\rm rms}$ of the same survey. WISE candidate host galaxy identified by RGZ DR1 is shown as a green ring, while possible SDSS host galaxies I found are shown in a black ring. The host galaxy position of J1646+3627 locates within the radio center of its VLASS/FIRST emission, which can be seen on the figure.

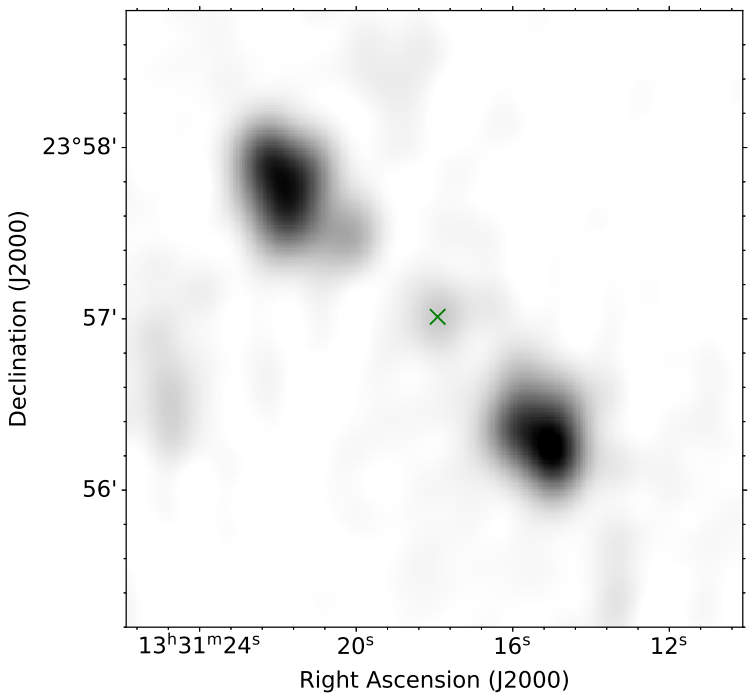


FIGURE 4.3: The processed greyscale image of J1331+2357 using the VLA public project AG0635 data, where a faint but visible core is seen at the center of the image. The green cross in the image indicates the host galaxy position given in Table 4.1.

GRG	W1	W2	W3
J0941+3126	15.165±0.038	14.650±0.062	11.595±0.204
J1331+2357	14.704±0.030	14.441±0.048	>12.205
J1402+2442	14.763±0.031	14.319±0.045	12.488±0.415
J1421+1016	15.104±0.033	14.703±0.054	12.841±0.512
J1646+3627	13.944±0.141	13.799±0.031	>12.275

TABLE 4.2: A summary of source infrared properties. WISE magnitudes in the table are extracted from the AllWISE catalogue (Cutri & et al., 2013) via **VizieR** (Ochsenbein et al., 2000).

GRG	VLSSr 73.8 MHz	TGSS ADR1 147.5 MHz	7C 151 MHz	WENSS 325 MHz	TXS 365 MHz	NVSS | FIRST 1.4 GHz	VLASS 3 GHz	MIT-Green 5 GHz	log $P_{1.4}$ [W/Hz]
J0941+3126	1160±202	925±68	930±76	302±87		219±26 | 144±2	126±1		25.73
J1331+2357	2747±676	1214±62	1270±149		901±105	177±22 | 100±1	143±1		25.81
J1402+2442	2728±541	1592±111	1840±140		812±73	381±41 | 263±4	120±1	137	26.15
J1421+1016	2078±495	884±67			701±75	242±30 | 184±4	98±1	106	26.05
J1646+3627	521±160	53±6		41±15		35±6 | 30±1	17±3		25.36

TABLE 4.3: A summary of source integrated flux densities, which are measured in mJy. Surveys including the VLA Low-frequency Sky Survey Redux (VLSSr; Lane et al., 2014), the NRAO VLA Sky Survey (NVSS; Condon et al., 1998), FIRST (Becker et al., 1995), and VLASS (Lacy et al., 2020) are done by the Very Large Array (VLA; Thompson et al., 1980). Source flux densities from the GMRT 150 MHz all-sky radio survey (TGSS; Intema et al., 2017), the Westerbork Northern Sky Survey at 325 MHz (WENSS; Rengelink et al., 1997) are also measured. I further found literature flux densities from the 7C survey of radio sources at 151 MHz (Waldram et al., 1996), the Texas Survey of Radio Sources at 365 MHz (Douglas et al., 1996) and the MIT-Green Bank Survey at 5 GHz (Bennett et al., 1986; Langston et al., 1990). Source flux densities are calibrated to a common flux scale of Scaife & Heald (2012). The radio luminosity $\log P_{1.4}$ is based on the NVSS images.

4.3 Analysis and Discussion

The overall occurrence of GRGs in the RGZ DR1 catalogue is 0.08%, which is slightly lower than that of LoTSS DR1. There are two potential reasons for this difference. Firstly, the RGZ citizen scientists are provided with only small-sized images to classify (3×3 arcmin^2), which limits the LAS of radio galaxies that can be fully contained in the image cutouts. Among the 11 237 galaxies considered in this work, the maximum source LAS is 195 arcsec. For GRGs to have angular sizes smaller than this requires them to lie at redshifts $z \geq 0.213$. Under a similar restriction, Dabhade et al. (2020a) would have missed 26.3% of their discovered GRGs. Correspondingly, the Kozieł-Wierzbowska et al. (2020b) and Kuźmicz et al. (2018b) samples would have missed as much as 62.5% and 66.4% of their catalogued GRGs within the sky area covered by RGZ DR1, respectively. Secondly, GRGs have historically been poorly detected in radio surveys like FIRST in part due to their synchrotron spectral index. The radio lobes of GRGs share relatively steep spectral indices, i.e. they are brighter at lower frequencies and thus in principle can more easily be found at MHz frequencies compared to GHz (Dabhade et al., 2020a).

In addition, finding GRGs in radio surveys like FIRST is limited by instrumental considerations. Interferometers with comparatively long baselines (as a function of wavelength) may not be sensitive to the large-scale emission associated with extended or diffuse radio sources (Saxena et al., 2018), nor may it always be encompassed by the comparatively small field-of-view for single-pixel centimetre-wave receivers. Such issues have in part been alleviated by radio telescopes such as the Expanded Very Large Array (EVLA; Sahr et al., 2002), and LOFAR at MHz-frequencies, and by telescopes with large instantaneous fields of view due to Phased Array Feed (PAF) technology, such as the Australian SKA Pathfinder (ASKAP; Johnston et al., 2008); however, whilst these instruments may be powerful probes of GRGs in the future (Peng et al., 2015) instrumental selection effects will always persist.

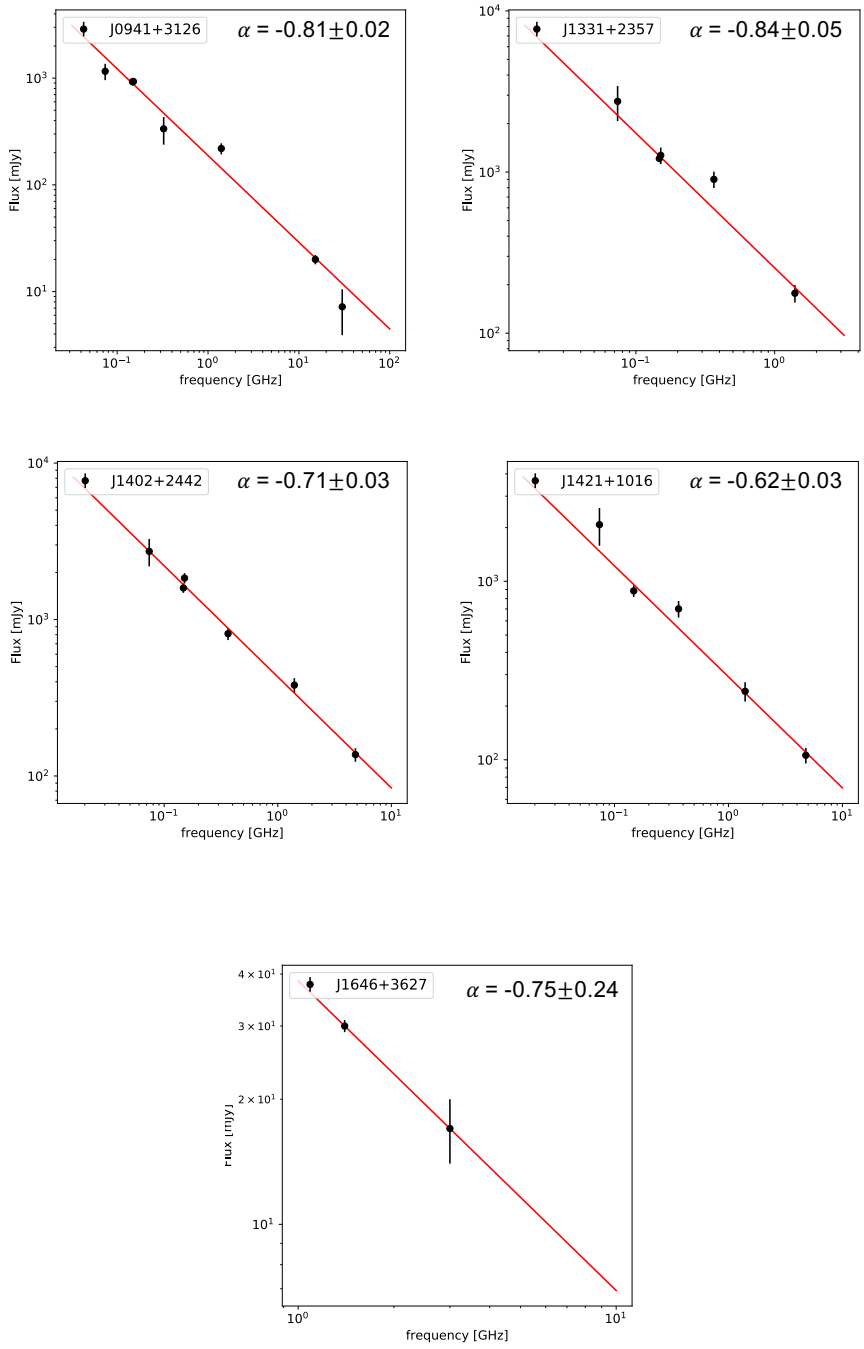


FIGURE 4.4: Continuum radio spectra of our GRGs. The solid red lines are linear least-squared fits, where the data points are weighted with their measurement errors when estimating the source spectral indices. Data points used for deriving source spectral indices are from Table 4.3 and Section 4.2. Considering angular resolution difference of the survey data, I adapted data from NVSS at 1.4 GHz and not using VLASS data for the top four sources. When deriving the spectral index of J1646+3627, I only consider FIRST and VLASS as they show clear radio core emission and have comparable angular resolution. Meanwhile, I didn't find clear visible radio core from ohter cited surveys..

Consideration of selection effects is of particular importance in the context of developing automated deep learning based GRG classifiers. Such algorithm development is complicated by a lack of large, uniform, and reliable cross-matched radio source catalogues that contain source information characterised in a consistent manner appropriate for the formation of computationally tractable training data. Furthermore, a key aspect of the development of potential machine learning based GRG classification algorithms, as well as radio galaxy classification more generally, is a clear understanding of the biases that are introduced by this training data selection. In this respect the RGZ DR1 catalogue represents a well-understood data sample where considerations such as input image size are pre-defined. Hence, although the restricted image size is considered a disadvantage for compiling large catalogues of GRGs, it is potentially an advantage for defining a deep learning training dataset with well understood data constraints.

4.4 Radio Source Luminosity

I measured the integrated flux densities for each source using images from the VLSSr, TGSS ADR1, WENSS, NVSS, FIRST and VLASS surveys. I also retrieved archival integrated source flux densities from the 7C, TXS, 9C, and MIT-Green Bank (MG) surveys using the NED database[1], these are listed in Table 4.3. At low frequencies, all historic data in Table 4.3 have been re-scaled to match the Scaife & Heald (2012) flux density scale, which is consistent with the Perley & Butler (2017) flux scale at higher frequencies. The five GRGs identified in this chapter have higher integrated flux densities in the NVSS survey (FWHM = 45″) than the FIRST survey (FWHM = 5.4″), which is consistent with a lack of shorter baseline coverage in the FIRST survey compared to that of NVSS. I note that the VLASS measurement should also be treated with caution as all five objects have diffuse emission on angular scales larger than 30″, which will be poorly recovered by this survey and result in underestimated integrated flux densities (Lacy et al., 2020).

The resulting spectra for all sources are shown in Figure 4.4. I find that these sources have a range of spectral indices from $-0.84 < \alpha < -0.62$, where $S \propto \nu^{\alpha}$, with an average spectral index of $\langle\alpha\rangle = -0.75$. This is similar to the mean spectral index, $\langle\alpha\rangle_{0.151}^{1.4} = -0.79$, found for the GRG sample of (Dabhade et al., 2020a) and is also consistent with the typical value for radio galaxies more generally (Kuźmicz et al., 2018b). Finally, our result happens to have the same view with Hardcastle et al. (2019b); Shabala et al. (2020) that these long-lived large radio galaxies are the tail of the radio galaxy age distribution.

Since FIRST and VLASS have comparable resolution and flux density loss on similar scales, I also compute the spectral index, $\alpha_{1.4}^{3.0}$, of the source core and lobes separately for J1646+3627 where the radio core is visible and has peak flux density above 3 σ_{rms} in both surveys. I found that the core region of the source has $S_{1.4} = 2.99 \pm 1.21$ mJy and $S_3 = 2.7 \pm 0.3$ mJy, giving a source core spectral index of $\alpha_{1.4}^{3.0} = -0.13$, and $\alpha_{1.4}^{3.0} < -0.69$

[1]NED is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

for the lobes. This is consistent with other resolved radio systems, where super-position of multiple synchrotron emission components results in a flatter core spectrum.

4.4.1 GRGs that are also BCGs

The GRG J1646+3627, newly identified here, is also the brightest cluster galaxy in the galaxy cluster GMBCG J251.67741 + 36.45295 (Hao et al., 2010). To better understand this emerging population, I performed a literature search for GRGs that are already known but have not previously been identified as a BCG. Following Dabhade et al. (2020a), I cross-matched the Kuźmicz et al. (2018b) catalogue with the GMBCG (Hao et al., 2010) and WHL (Wen et al., 2012) galaxy cluster catalogues.

This search returned 13 new BCG GRG candidates, which are listed in Table 4.4. 10 out of 13 of these candidates have not been identified as BCGs previously due to a historic lack of availability of large scale optical galaxy cluster catalogues when they are discovered. The other 3 candidates are not recognized as finding BCG GRGs was not highlighted in Proctor (2016).

Prior to this work, 21 BCG GRGs were identified by Dabhade et al. (2017, 2020a). Combining that sample with the 13 I identify from the literature and the one new GRG BCG from RGZ DR1, there are 35 BCG GRGs known in total. From the full sample of 35 BCG GRGs, 28 are in clusters with catalogued R_{200}, the radius where the mean density is at least 200 times the critical density of the Universe, and N_{200}, the local galaxy number within R_{200} (Wen et al., 2012). N_{200} here only counts galaxies with $M_e^r(z) \leq -20.5$, where $M_e^r(z)$ refers to evolution-corrected absolute magnitude in the *r* band (Wen et al., 2012): $M_e^r(z) = M_r(z) + Qz$. A passive evolution of Q=1.62 was adopted (Blanton et al., 2003). Using these data, the relationship between local galaxy density and projected linear size for these galaxies is shown in Figure 4.5. Of these 28 objects, there are five with host galaxy redshifts $0.05 < z < 0.15$, the same range used by Malarecki et al. (2015) who also investigated the relationship between GRG size and local environment. I re-calculate the galaxy number density of each cluster for these five objects assuming a cylindrical volume with a radius of R_{200} for each source. Consistent with Malarecki et al. (2015), I adopt a physical cylinder length equivalent to $z = 0.1 \pm 0.003$. This returns galaxy number density values from 0.11 to 0.27 Mpc^{-3} with a median galaxy number density of 0.24 Mpc^{-3}. These are shown as yellow data points in Figure 4.5. Original data from Malarecki et al. (2015) are shown as blue data points; however, the values of R_{200} for these galaxies are generally closer to 1 Mpc than the cylinder radius of 2 Mpc used by Malarecki et al., consequently I also show the local galaxy density for the sources in Malarecki et al. (2015) re-calculated using a cylinder radius of 1 Mpc. These data are shown as red points in Figure 4.5. The maximum galaxy number density of the 23 BCG GRGs with with host galaxy redshifts $z > 0.15$ under the same volume assumption is 0.38 Mpc^{-3}.

From Figure 4.5 it can be seen that the BCG GRGs have been growing in generally denser environments than the non-cluster/poor cluster GRGs in the sample of Malarecki et al. (2015). When considering a radius of 1 Mpc, source B 1308-441 from the Malarecki et al. (2015) sample has a comparable local galaxy number density to the BCG GRGs, due

GRG ID	Cluster ID	RA (J2000.0) [h:m:s]	DEC (J2000.0) [°:′:″]	z	R_{200} [Mpc]	N_{200}	R_{L*}	M_{200} [$10^{14}M_{\odot}$]
J1054+0227	GMBCG J163.58817+02.46528	10:54:21.16	+02:27:55.0	0.34	–	–	–	–
J1400+3019	GMBCG J210.18097+30.32185	14:00:43.43	+30:19:18.7	0.206	–	–	–	–
J0115+2507	WHL J011557.2+250720	01:15:57.23	+25:07:21.0	0.1836	0.96	15	18.28	1.0
J0129−0758	WHL J012935.3−075804	01:29:35.26	−07:58:04.3	0.0991[a]	1.17	10	28.44	1.6
J0751+4231	WHL J075108.8+423124	07:51:08.80	+42:31:24.2	0.2042	0.98	14	17.87	0.9
J0902+1737	WHL J090238.4+173751	09:02:38.42	+17:37:51.5	0.1645[a]	1.01	14	19.68	1.1
J0926+6519	WHL J092600.8+651923	09:26:00.82	+65:19:22.7	0.1397	0.84	8	14.41	0.7
J1108+0202	WHL J110845.5+020241	11:08:45.49	+02:02:40.9	0.1574[a]	1.05	26	23.55	1.3
J1235+2120	WHL J123526.7+212035	12:35:26.67	+21:20:34.8	0.4227	0.79	10	12.03	0.6
J1418+3746	WHL J141837.7+374625	14:18:37.65	+37:46:24.5	0.1349	1.17	25	28.14	1.6
J1453+3308	WHL J145302.9+330842	14:53:02.86	+33:08:42.4	0.2482	0.92	14	16.69	0.9
J1511+0751	WHL J151100.0+075150	15:11:00.01	+07:51:50.0	0.4594	1.09	17	23.20	1.3
J2306−0930	WHL J230632.2−093020	23:06:32.18	−09:30:20.6	0.1593	1.03	16	20.35	1.1

TABLE 4.4: A summary of the BCG GRG candidates I found from Kuźmicz et al. (2018b). RA/DEC, redshift, FR type, and Reference number are extracted from Kuźmicz et al. (2018b). The galaxy cluster ID are extracted from GMBCG (Hao et al., 2010) and WHL (Wen et al., 2012) galaxy cluster catalogues. R_{200}: the radius of a cluster that its mean density is 200 times of the critical density of the universe; N_{200}: the galaxy number within the R_{200}; R_{L*}: cluster richness; M_{200}: the mass of a cluster that its mean density is 200 times of the critical density of the universe, which is derived from R_{L*} using the Equation 2 of Wen et al. (2012). **References**: 1. Baum & Heckman (1989), 2. Best et al. (2005), 3. Lara et al. (2001b), 4. Machalski et al. (2007), 5. Nilsson (1998), 6. Parma et al. (1996), 7. Proctor (2016), 8. Schoenmakers et al. (2001). [a]: The cluster redshift and the source redshift have a difference of $0.03 - 0.04$, the cluster membership of these radio sources should be treated which caution.

to a concentration of galaxies in close proximity. The mean galaxy number density of the Malarecki et al. (2015) GRGs using a cylinder radius of 1 Mpc is $0.07\,\mathrm{Mpc}^{-3}$, typical for a poor cluster or galaxy group.

For the BCG GRGs I also compute the cluster mass, M_{200}, in each case. With the exception of WHL J112126.4+534457 with a mass of $4.6\times10^{14}M_{\odot}$, the masses of these clusters lie in the range $0.7 - 2 \times 10^{14}\,M_{\odot}$. Given that the average M_{200} for the WHL catalogue is $\sim1.12\times10^{14}\,M_{\odot}$, the masses of these particular clusters are unremarkable with respect to the wider catalogue.

The existence of these BCG GRGs, though they only represent a small population of GRGs, implies that sparse environment itself can hardly be the only factor to explain how GRGs grow into Mpc scales, though the probability of finding BCG as a GRG is still low. Another recent study done by Dabhade et al. (2020c) had further supported this idea, where they had identified 60 BCGs out of a sample of 820 GRGs (of which 162 GRGs are newly discovered). Their investigation upon the 60 BCG GRGs claimed that local environment does play a role in the growth of GRGs, while it is not the only factor that affects their sizes (Dabhade et al., 2020c).

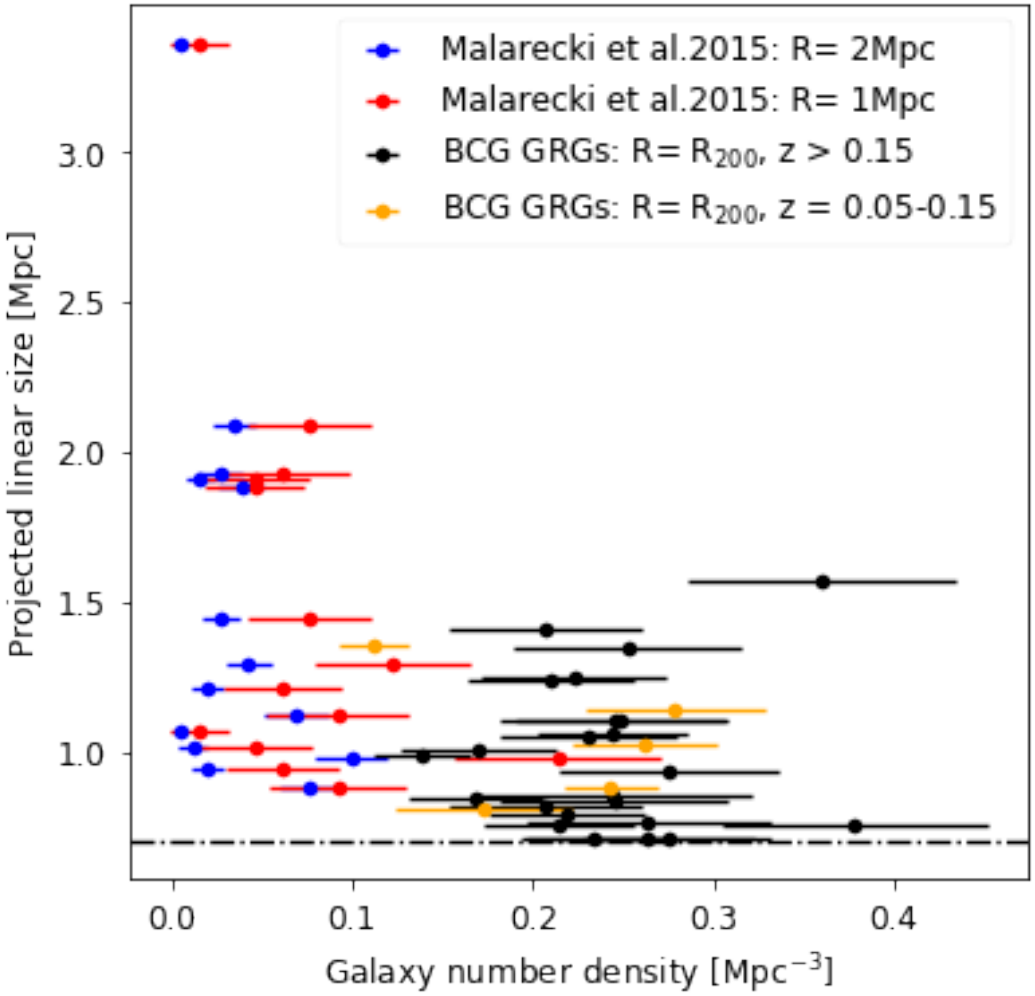


FIGURE 4.5: A diagram of galaxy number density vs. source projected linear size, comparing samples discussed in Malarecki et al. (2015) and BCG GRGs with R_{200} and N_{200} available from the WHL catalogue. R_{200} and N_{200} of each BCG GRGs in the diagram can be found in Table 4.4 and the Table 3 of Dabhade et al. (2020a). The galaxy number density uncertainty of the BCG GRGs are estimated based on the Equation 1 of Wen et al. (2012) and our cylindrical volume assumption. The galaxy number density uncertainty of Malarecki et al. (2015) samples are extracted from the Table 4 of their work. The dashed line in the diagram equals 700 kpc.

4.5 Conclusion

In this work I have identified 5 new GRGs from RGZ DR1. These GRGs mostly share an FR II radio morphology and cover the redshift range of $0.28 < z < 0.43$. These GRGs have been identified using a method consistent with the assembly of training data appropriate for a deep learning classifier. I compare the selection of these GRGs to previous studies and suggest that samples defined in this manner are more likely to be representative of future deep learning approaches to GRG identification than previous methods.

I associate one of the newly identified GRGs with the brightest cluster galaxy in galaxy cluster GMBCG J251.67741+36.45295 (Hao et al., 2010) and using literature data I identify a further 13 previously known GRGs to be BCG candidates. This increases the number of known BCG GRGs by more than 60%. I show that the local galaxy density of these sources is significantly higher than that of non-cluster GRGs, challenging the hypothesis that giant radio galaxies are able to grow to such large sizes only due to locally under-dense environments.

Chapter 5

Branched CNNs for GRG classification

The work in this chapter is in preparation for submission to the *Monthly Notices of the Royal Astronomical Society*.

Giant Radio Galaxies (GRGs) are generally rare among radio galaxies. These objects are usually seen as probes of WHIM, where WHIM are thought to be the place that the missing baryons outside the galaxies reside (Peng et al., 2015). In the last 47 years of GRG study, though people have found that these objects might have higher Eddington ratio compared to those RGs of smaller size, a few GRG related questions remain unsolved (Dabhade et al., 2020c): How do these objects form? How rare are they? Do they host the most powerful Super Massive Black Hole (SMBH; Soltan, 1982)? In order to probe WHIM in the universe, so as to investigate these problems, it is then necessary to continue the work of GRG hunting.

The identification of GRGs, however, has conventionally relied on the measurement of source LAS and their host galaxy redshifts. Sample LAS, especially, require researchers to have a clear definition of source radio component boundaries. So far it is yet to be discovered whether an automated algorithm could identify the boundaries of a sample of object radio components and estimate its LAS solely based on the sample image data. It is also interesting to investigate whether such an algorithm could identify a giant with or without the facilitation of host galaxy redshift information. The two problems have motivated me to perform study upon possible algorithm development in the following chapter.

In this chapter I introduce a novel automated method for classifying radio galaxies as giants or non-giants solely based on a convolutional neural network approach. The structure of the chapter is as follows: in Section 5.1 I describe how I selected and built two machine learning data sets with different object class ratios and sample constitutions; in Section 5.2 and Section 5.3 I compare the model performances resulting from different training strategies and data samples, including an interpretation of different model behaviour and the connection to data selection; I finally draw our conclusions in Section 5.4.

In this work I assume a ΛCDM cosmology with $\Omega_m = 0.31$ and a Hubble constant of $H_0 = 67.8$ km s^{-1} Mpc^{-1} (Planck Collaboration et al., 2016). The model training, validation and testing are performed using the Google Colaboratory (Bisong, 2019), which is equipped with an NVIDIA Tesla T4 (14.7 GB memory).

5.1 GRGNOM: Dataset Construction

Both citizen science and traditional astronomy methods are now being used as the foundation for recent studies that employ automatic radio morphology classification using deep learning algorithms. Using a number of radio galaxy catalogues which include radio morphology classification, e.g. FRICAT; Capetti et al. (2017a), FRIICAT; Capetti et al. (2017b), the Combined NVSS-FIRST Galaxies (CoNFIG) sample; Gendre & Wall (2008); Gendre et al. (2010) and Mingo et al. (2019), several automatic radio morphology deep learning classifiers have been developed (e.g. Aniyan & Thorat, 2017; Alhassan et al., 2018; Lukic et al., 2018, 2019; Ma et al., 2019). These automatic classifiers are built to extract morphological features from input images for classification. For general radio galaxy classification (Fanaroff & Riley, 1974), these applications have achieved model accuracies comparable to visual inspection. However, these deep learning algorithms require their individual image inputs either to have a common input image size or to be resized to a common size (Lukic et al., 2019). Since GRG identification requires LAS estimation, training a deep learning based GRG classifier under these constraints would require an image training dataset with image sizes large enough to enable an algorithm to estimate source LAS for very extended objects. Considering the memory limits of state-of-the-art GPUs, the image sizes required for such an algorithm in the GRG case are likely to make such a general approach highly computationally expensive. Consequently, in the case where image size is restricted due to memory limitations, as well as the potential for confusion due to multiple objects in the field, careful consideration must be given to the effects of selection bias in the use of such machine learning approaches.

5.1.1 General Guideline

Ideally, the GRG classifiers I intend to build should be able to find differences in physical linear size, rather than only be able to distinguish source angular extent differences. It is therefore necessary for us to build a data set including confirmed GRGs and radio galaxies with smaller sizes. Also, given that GRG hunting would usually consider multi-frequency/resolution image data, the data sample selection should also follow the rules of data set foundation for training Multi-branched CNNs. Consequently, I selected data samples and built our data sets according to the following criteria:

Data availability: All data samples should include (i) image survey data from both the NVSS and FIRST surveys, (ii) host galaxy redshifts, (iii) Largest Angular Size (LAS) measurements, and (iv) have a physical linear size calculated.

Source-Image relationship: Image data from each radio survey should have the same image size in terms of angular size.

Image Pre-processing: After pre-processing images should only contain positive-valued pixels and the source should be visible in the image.

Traceability: Users should be able to trace the source coordinates, catalogued object ID, and original source catalogue for each sample.

I require data traceability when building our data set not only as it will benefit model training and testing, but also it allows both users and developers to evaluate and explain model outcomes based on their scientific understanding of the data. For example, in this work, I make use of the data traceability in Section 5.3 to explain mis-classifications with respect to different model architectures.

5.1.2 Data Sample Selection

Radio galaxies of smaller sizes

Radio galaxies with non-giant dimensions were selected from Data Release 1 of Radio Galaxy Zoo (RGZ DR1; Wong et al. in prep.). RGZ DR1 is a radio galaxy catalogue created by over 12 000 volunteers through the RGZ citizen science project. Project users are asked to cross match radio source lobes with a corresponiding infrared host galaxy. The radio images come mainly from the FIRST survey, and the infrared images are mainly 3.4 μm $WISE$ images. These radio and infra-red images share a 3×3 arcmin field of view. RGZ DR1 is a summary of the first 2.75 years of cross-matches since the project launched, during which time it has returned over 75 641 identifications (Wong et al. in prep.).

A previous investigation of the DR1 catalogue has shown that the uniform 3×3 arcmin image size constrains its ability to identify GRGs (Tang et al., 2020). However, the catalogue does provide a large source sample with full data availability and traceability, as required for the construction of a machine learning dataset. Previous analysis of the full RGZ DR1 catalogue found that at least 11 237 non-duplicated samples fulfill the requirements for training data set selection outlined at the start of Section 5.1.1. These samples have LAS from 1.6 arcsec to 195.4 arcsec, and are all within the 3×3 arcmin field of view.

In addition to the source-image relationship required for each data set sample, I further require that the radio centroid and host galaxy position should be generally consistent for each galaxy. This is because the estimated radio centroid in the RGZ DR1 catalogues is not necessarily consistent with the position of the host galaxy (Wong et al. in prep.). Consequently, I only retain sources with an angular separation between the host galaxy and estimated radio centre smaller than 1.8 arcsec, which is the pixel size of FIRST survey images (Becker et al., 1995). This criterion reduces the 11 237 RGZ DR1 samples to 6 021 samples. On inspection, I also removed the known GRG source GRG J1402+2442, which was included in the RGZ DR1 sample. Finally, I examined the survey image data availability of NVSS and FIRST using the SkyView API query

`astroquery.skyview.get_image_list`, to confirm that all sources had the required image data.

Giant Radio Galaxies

The GRG sample for this work comes from the Kuźmicz et al. (2018a) and Dabhade et al. (2020b) catalogues. Kuźmicz et al. (2018a) performed a detailed review of the literature identifying 349 GRGs, of which 89.7% are FR II objects. The catalogue has its sources validated using the NVSS at 1.4 GHz and measures their flux densities using those data. However I note that primary image data used for GRG identification in the literature covers a wide range of image angular resolutions, from arcsec level to 45″ (e.g. LoTSS, NVSS, SUMSS).

Dabhade et al. (2020b), on the other hand, performed an independent GRG search using the Value Added Catalogue (VAC; Williams et al., 2019) of the LOw Frequency ARray (LOFAR; van Haarlem et al., 2013). The team identified 239 GRGs, including 225 which were previously unknown. The newly discovered GRGs in the catalogue of Dabhade et al. (2020b) have their candidates identified from LoTSS survey images at 151 MHz with 6″ resolution. They cross-validated their sources using the FIRST, WENSS and TGSS surveys (Dabhade et al., 2020b). The difference in GRG sample selection and cross-validation between these samples will allow us to compare model behaviour when using samples selected with different class definitions. I discuss this further in Section 5.3.2.

From these catalogues I found that 310 GRGs in the Kuźmicz et al. (2018a) catalogue have NVSS images available, and 186 also have FIRST images. All newly discovered GRGs in the Dabhade et al. (2020b) sample have both NVSS and FIRST images available. Considering that the maximum source LAS in the RGZ DR1 entries is 195.4 arcsec, I further require the GRG samples to have LAS equal or smaller than the DR1 LAS limit. This reduces the samples to 58 GRGs from the Kuźmicz et al. (2018a) catalogue and 167 GRGs from the Dabhade et al. (2020b) catalogue.

5.1.3 Image pre-processing and further sample selection

In this work all image data are obtained using the Skyview Virtual Observatory[1]. Consistent with the RGZ DR1, I define our image data to have a field of view with a uniform 3×3 arcmin size. This is equivalent to 100×100 pixels for the FIRST images where 1 pixel = 1.8″, and 18×18 pixels for the NVSS images where 1 pixel = 15″. I acquire both the NVSS and FIRST postage stamp images in FITS format, defining the image centres using the host galaxy position from RGZ DR1.

The original FITS images are linearly scaled, and have units of Jy/beam. Following the literature, I then subject each FITS image to a series of pre-processing steps before use. Aniyan & Thorat (2017) highlighted the importance of image noise reduction, and replaced all image pixels with values lower than a specified noise threshold with zeros, giving the sample images cleaner backgrounds and enabling neural networks to train

[1] https://skyview.gsfc.nasa.gov

with high sparsity. Specifically, they proposed this sigma-clipping for each sample image to be at the 3-$\sigma_{\rm rms}$ level. Later studies showed that applying sigma-clipping at a 3, 4 or 5-$\sigma_{\rm rms}$ level did not significantly change the model outcome, and the approach of sigma-clipping generally has been applied successfully by a number of applications (e.g. Aniyan & Thorat, 2017; Tang et al., 2019).

In this work, I also follow the method of Aniyan & Thorat (2017) and sigma-clip the images individually at a level of $3\sigma_{\rm rms}$. I then apply the following image normalization to each of the images:

$$\text{Output} = \frac{\text{Input} - \text{Min}}{\text{Max} - \text{Min}} \times (255.0 - 0.0). \tag{5.1}$$

Following this pre-processing, I found that some radio sources with very low signal to noise ratios were eliminated. I found that 15, 7, and 15 objects from the DR1, Kuzmicz and Dabhade source catalogues, respectively, had at least one of the FIRST or NVSS images result in an empty field of view and consequently these objects were removed from our sample.

Figure 5.1 shows the size distribution of the remaining objects, where the red dashed line indicates a source size of 500 kpc and the purple line represents 700 kpc, the GRG size cut-off. Only four objects have linear sizes intermediate to these two values and, for clarity, I exclude these four intermediate sources from the data set and define the two target classes of radio galaxy in this work to be:

- NOM: Radio galaxies with linear sizes smaller than 500 kpc.
- GRG: Radio galaxies with linear sizes larger than 700 kpc, consistent with the standard definition from the literature.

Following the pre-processing described above, the sample contains 6001 radio galaxies of class NOM, and 205 of class GRG. Since the two classes have a clear difference in linear size, in principle a good classifier should be able to distinguish them well. All of these objects are centred in their images. Data for each sample also includes source object ID, qualified host galaxy redshift, LAS measurement, and computed source linear size. However, as described in Section 5.2 later in this work, only image data and redshift information are used as model inputs for the classifiers evaluated in this work.

5.1.4 Data formating, division and summary

Although there are $\sim$ 6000 sample sources of class NOM, the dataset would be extremely imbalanced if I use all of them when building our training/testing set. In consideration of the observationally imbalanced nature of GRGs and sources with smaller sizes, I use these data to build a modestly imbalanced dataset of 600 training samples and 200 testing samples, with GRG samples extracted from the Kuźmicz et al. (2018a) catalogue only, and samples of class NOM taken from RGZ DR1. The resulting dataset is named GRGNOM-A, and has a class balance of $\sim 14:1$. Given the availability of the Dabhade

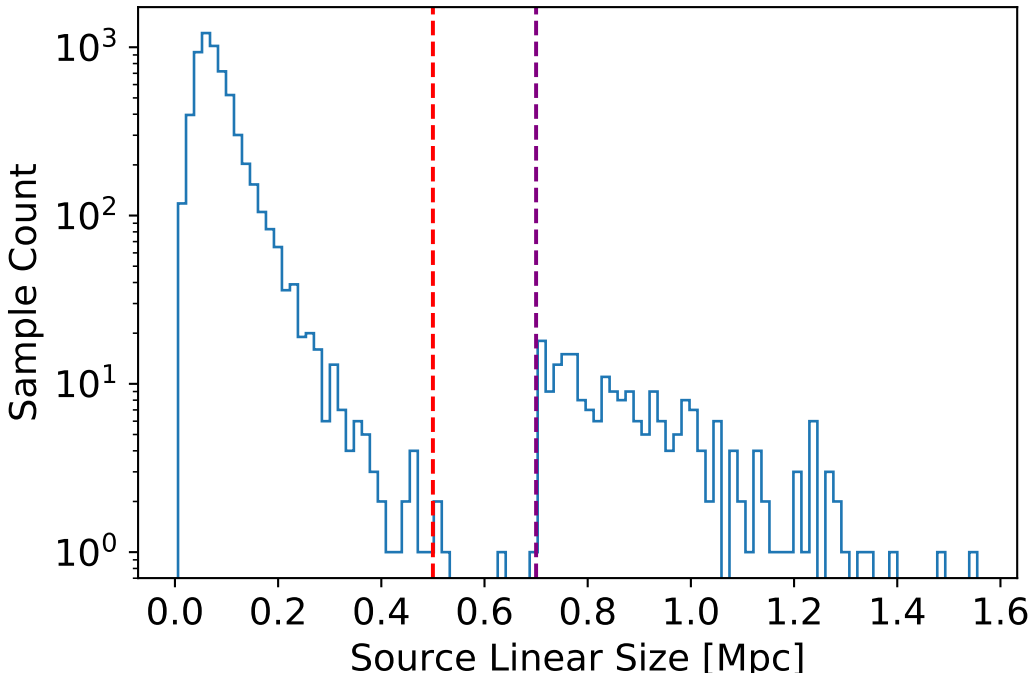


FIGURE 5.1: Bulk sample projected linear size distribution after I performed sample section procedure in Section 5.1.2 and Section 5.1.3. The red and purple dashed lines on the diagram indicate projected linear sizes of 500 kpc and 700 kpc, respectively.

et al. (2020b) samples, I also build a second dataset using 204 out of 205 GRGs in the selected data samples, which I refer to as GRGNOM-B. This data set has a NOM:GRG ratio of approximately 3 : 1 in both the training and test data samples. Given that the test set of GRGNOM-A only contains Kuźmicz et al. (2018a) samples, I only include Dabhade et al. (2020b) samples in the GRGNOM-B test set in order to allow models trained using either data set to test their generalization ability. However, since a large number (115) of the class NOM objects between GRGNOM-A and GRGNOM-B testing are overlapped. I decided to build a generalization test these models with another 149 RGZ DR1 samples of class NOM that are not found in either GRGNOM-A or GRGNOM-B. I refer to the resulting test set as GRGNOM-Gen. The resulting data samples are summarized in Table 5.1.

It can be seen that the GRGNOM-B training sample is dominated by Dabhade et al. (2020b) samples, while the GRGNOM-A data set only includes Kuźmicz et al. (2018a) samples. In particular, the test set of GRGNOM-B only contains samples from the Dabhade et al. (2020b) catalogue, in order to see if the features learned from Kuźmicz et al. (2018a) samples can facilitate the identification of Dabhade et al. (2020b) samples. The data sample construction also gives those models trained with GRGNOM-B a chance to test their generalization ability upon samples with their identification, source LAS, and host galaxy redshifts measured in a uniform manner.

In terms of data format, both GRGNOM-A and GRGNOM-B are split into two parts: (i) text-format tables of numerical source information, e.g. Table 5.2, and (ii) a group of

GRGNOM-A	NOM	GRG		Total
	RGZ DR1	Kuzmicz	Dabhade	
Training	561	39	0	600
Testing	187	13	0	200
Count	748	52	0	800
GRGNOM-B	**NOM**	**GRG**		Total
	RGZ DR1	**Kuzmicz**	**Dabhade**	
Training	447	52	101	600
Testing	149	0	51	200
Count	596	52	152	800
GRGNOM-Gen	**NOM**	**GRG**		Total
	RGZ DR1	**Kuzmicz**	**Dabhade**	
Testing	149	0	51	200
Count	149	0	51	200

TABLE 5.1: A summary of the sample division of the GRGNOM-A, GRGNOM-B and GRGNOM-Gen. 'Count' refers to the total source sample number of a class, and 'Total' represents the sample number of each column. The GRG samples in the GRGNOM-B testing set are the same as that of GRGNOM-Gen.

machine readable documents containing feature data in various formats. This second component of the data set is comprised of:

1. **FIRST images**: The pre-processed FIRST survey greyscale images with a universal size of 100×100 pixels. Three versions of these image files are generated:

   a. Image batch files: Images are saved in 4 batched files (3 training batches and 1 testing batch), along with a metadata file containing corresponding image header information. These files are in a format that is understandable for our `Pytorch` models. These files are saved in a folder named FIRST.

   b. Encoded compressed file: This is the encoded compressed file of **a**. The compressed file is named as FIRST.tar.gz. Creating such a data file allows future developers to download and re-use the image data samples for machine learning training.

   c. Individual images: These images are saved in another image folder named *FIRST IMG*. Image names follow the format: *CatalogueName_CatalogueSourceNo_RightAscension_Declination.png*.

2. **NVSS images**: The pre-processed NVSS survey greyscale images with a universal size of 18×18 pixels are saved in the same manner as the FIRST images.

3. **Source host galaxy redshift**: Consistent with the image data, numerical source host galaxy redshifts are separated into 4 batches and saved as `numpy` arrays.

4. **Source LAS**: Numerical source LAS are organized and saved in the same way as (iii).

Object ID	RA (J2000.0) [h:m:s]	DEC (J2000.0) [d:m:s]	z	LAS [arcsec]	Size [Mpc]	Label
RGZJ000606.0+013125	00:06:06.07	01:31:25.20	0.23372	21	0.079	NOM
RGZJ000626.4+081838	00:06:26.41	08:18:38.49	0.41540	18	0.102	NOM
RGZJ000627.2+060407	00:06:27.21	06:04:07.29	0.30091	14	0.064	NOM
RGZJ000746.4+031938	00:07:46.46	03:19:38.99	0.29194	44	0.200	NOM
RGZJ000851.0+045243	00:08:51.01	04:52:43.92	0.34255	19	0.095	NOM
RGZJ000911.0+145105	00:09:11.05	14:51:05.07	0.36832	20	0.105	NOM
RGZJ001042.9+091917	00:10:42.92	09:19:17.44	0.15308	19	0.052	NOM
RGZJ001051.0+141655	00:10:51.09	14:16:55.86	0.31507	17	0.079	NOM
RGZJ001146.7+101528	00:11:46.75	10:15:28.45	0.22175	17	0.064	NOM
RGZJ001524.2+143038	00:15:24.23	14:30:38.83	0.22668	33	0.122	NOM

TABLE 5.2: The first 10 rows of the GRGNOM-A training sample catalogue. Source object ID, RA/DEC, host galaxy redshift (z) and LAS are extracted from RGZ DR1, while the source linear size is derived from the z and LAS of each sample based on the cosmological parameters defined in Planck Collaboration et al. (2016). Class labels are defined as described in Section 5.1.3.

5. **Source linear size**: Numerical source linear size are organized and saved in the same way as (iii).

6. **Source object ID**: Primary object ID of each source sample in their original catalogues. These data strings have been encoded in the `uint8` format and saved in the same way as (iii). As necessary, they can be decoded in the `utf-8` format.

7. **Class label**: Numerical class label of each data sample: 0 and 1 represent source classes NOM and GRG, respectively. They are organized and saved in the same way as (iii).

By using the Python `pickle` package, I built our training/testing datasets with these components, which allow users to call any of the source data samples in the dataset. Hash values were generated separately for (i)a, (i)b, (ii)a and (ii)b to protect their integrity and avoid manipulation.

I note that although source LAS and physical size are included in the data set for completeness, this information is not used to train any of the models in this work.

5.1.5 Data Normalization and Augmentation

In order to improve the convergence of model training, it is recommended to define a data normalization and augmentation strategy (LeCun et al., 2012). Normalization constrains data values within a given range, reduces skew and therefore speeds up the training process of a model. In this work, I require the image data to be normalized to have both a mean and standard deviation of 0.5 before importing to a model, which constrains image pixel values largely within the range from 0 to 1.

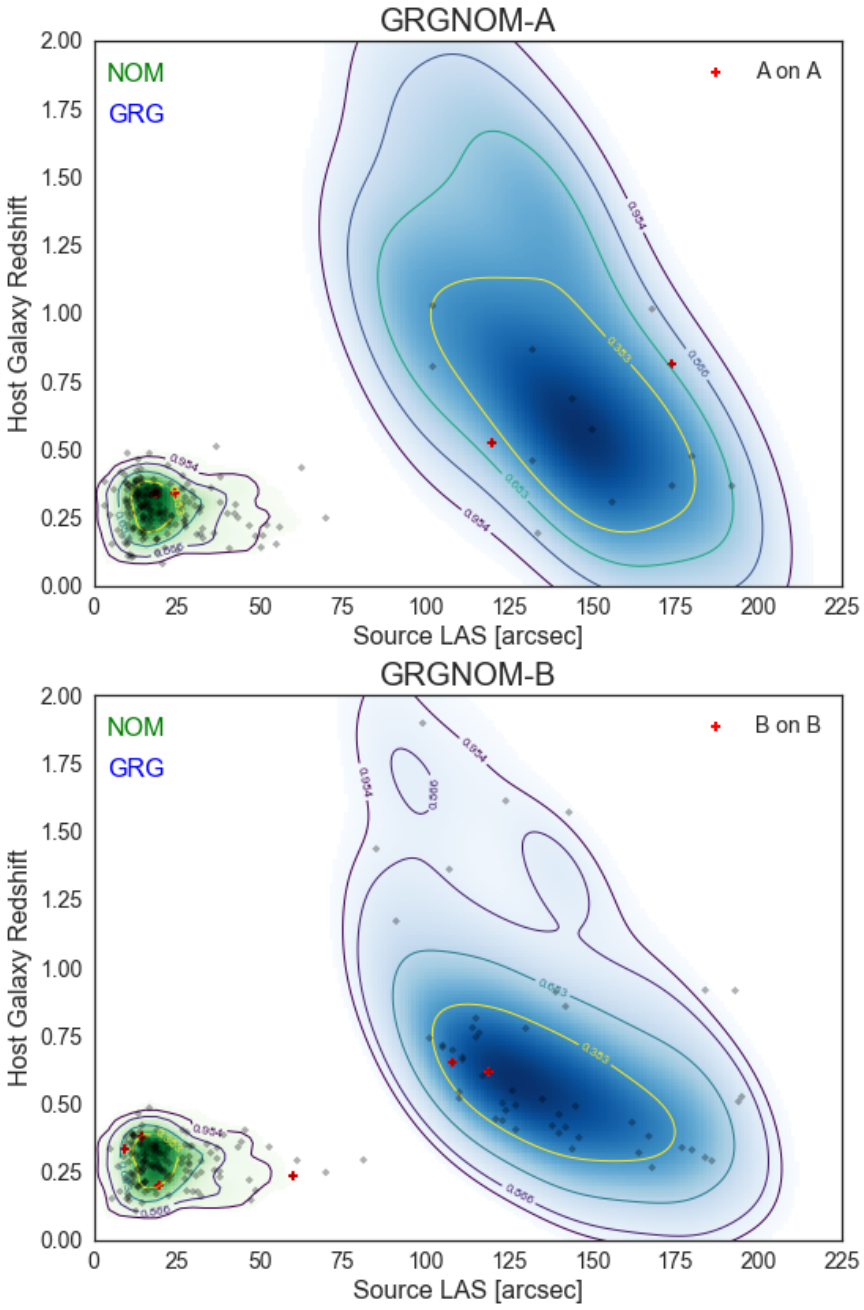


FIGURE 5.2: Upper: the LAS vs. host galaxy redshift density map of the GRGNOM-A dataset. Training samples of class NOM and GRG are represented in green and blue, respectively. Contours on the diagram refer to the iso-proportion of the density, i.e. 95.4% of the probability mass lies within the 0.954 contour. Contours are shown at 0.383, 0.683, 0.866 and 0.954 in the figure, which correspond to 0.5, 1, 1.5 and 2σ. Grey data points are the GRGNOM-A test sample data points and red data points on the diagram indicate samples that are frequently mis-classified by models of Architecture G trained and tested on the GRGNOM-A data set. Lower: equivalent distributions for the GRGNOM-B dataset. The red data points indicate samples that are frequently mis-classified using Architecture G trained and tested on the GRGNOM-B data set. This diagram was plotted using the `pyrolite` package (Williams et al., 2020).

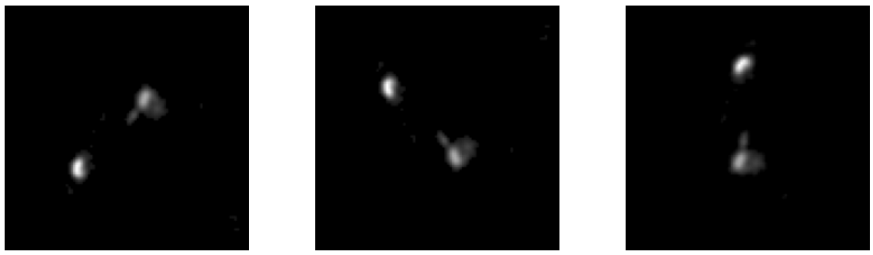


FIGURE 5.3: An illustration of image pre-processing and data augmentation using an example FIRST survey image (object id: Dabhade201), which is a radio source of class GRG with LAS of 108 arcsec.

In the case of data augmentation, I apply horizontal flipping and image rotation in this work (i.e.Figure 5.3). Specifically, I perform horizontal flipping of every image sample with a random probability of 50%. I then further randomly rotate each image in a clockwise manner using a randomly selected angle from -45° to 45°, where the rotated angle should be an integer in units of degrees.

Data augmentation is performed dynamically during training as the data are imported to the model. This strategy both ensures the model has a large enough number of data samples to learn and minimises memory usage. Using this approach, the training data set would have a statistical size of $600 \times 2 \times 360 = 432\,000$ samples when the model gets trained for 720 epochs.

5.2 Network Architecture

In this work I consider five different network architectures to create seven different models. These are summarised in Table 5.4. The first network architecture is a traditional, or classical, CNN approach that takes a single source of image data as an input, this forms the basis for Architecture A (NVSS) and Architecture B (FIRST) in Table 5.4. The second form of network is a multi-domain architecture that takes a single source of image data plus redshift information as its inputs, this forms the basis for Architecture C (NVSS + z) and Architecture D (FIRST + z) in Table 5.4. The third form of network is a multi-domain network that takes multiple sources of image data as inputs, this forms the basis for Architecture E (NVSS + FIRST), and the fourth form of network is the expansion of this architecture to include redshift as an input, Architecture F (NVSS + FIRST + z). The final form of network is Architecture G in Table 5.4, which has the same inputs as Architecture F but replaces all convolutional layers with Inception Modules, see Section 5.2.3.

For the convenience of parallel model performance comparison, any model training I performed in this work would uniformly use the Stochastic Gradient Descent optimizer (SGD; Robbins & Monro, 1951) for optimization. Training sample data would be imported to a model with batch size of 20, and the sequence of importing training data

Layer No.	Layer Type	Input Channels	Output Channels	Kernel Size	Stride	Activation	Regularization
1	Convolutional	1	6	5	1	ReLU	**IC**
2	Max Pooling	6	6	2	2		
3	Convolutional	6	16	5	1	ReLU	**IC**
4	Max Pooling	16	16	2	2		
5	Convolutional	16	120	5	1	ReLU	**IC**
5′	Squeeze layer 5 outputs						
6	Fully-connected	120×Down-sampled neuron number	120			ReLU	Dropout
7	Fully-connected	120	84			ReLU	Dropout
8	Fully-connected	84	2			Softmax	

TABLE 5.3: A summary of the modified LeNet-5 architecture used in this work as a base architecture. IC refers to the independent component (Chen et al., 2019).

Architecture	A	B	C	D	E	F	G
Input data	**NVSS**	**FIRST**	**NVSS & z**	**FIRST & z**	**NVSS & FIRST**	**NVSS & FIRST & z**	**NVSS & FIRST & z**
Convolution Branches	1	1	1	1	2	2	2
Layer 6 input (**NVSS**)	$120 \times 9 \times 9$		$120 \times 9 \times 9$		$120 \times 9 \times 9$	$120 \times 9 \times 9$	$128 \times 9 \times 9$
Layer 6 input (**FIRST**)		$120 \times 25 \times 25$		$120 \times 25 \times 25$	$120 \times 25 \times 25$	$120 \times 25 \times 25$	$128 \times 25 \times 25$
Layer 7 input	120	120	120 + 1	120 + 1	120 + 120	120 + 120 +1	120 + 120 + 1
Extra FC	No	No	No	No	Yes	Yes	Yes
Branch Module	Nil	Nil	Nil	Nil	Nil	Nil	Inception Modules

TABLE 5.4: The summary of architectures I adopted in this work. *Convolution Branches* refers to the number of independent top-down architectures from Layer 1 to Layer 5′ in Table 5.3.

would be shuffled every training epoch. In the context of other model hyper-parameters selection, I performed hyper-parameter grid searching on model architecture A, B and E, along with different initial learning rates (1e-3, 1e-4, 1e-5), dropout rates (0.4, 0.5, 0.6) and training epoch numbers (360, 720, 1080) upon both GRGNOM-A and GRGNOM-B datasets. In order to both prevent resulting models from over-fitting and achieve best model performances (i.e. model accuracy, AUC value), I decided to apply initial learning rate of 1e-3 and dropout rate of 0.4. I further apply early-stopping at epoch 1080/360 for models trained and validated with the GRGNOM-A/B data set. In the rest of this section, I describe the components of these networks in more detail.

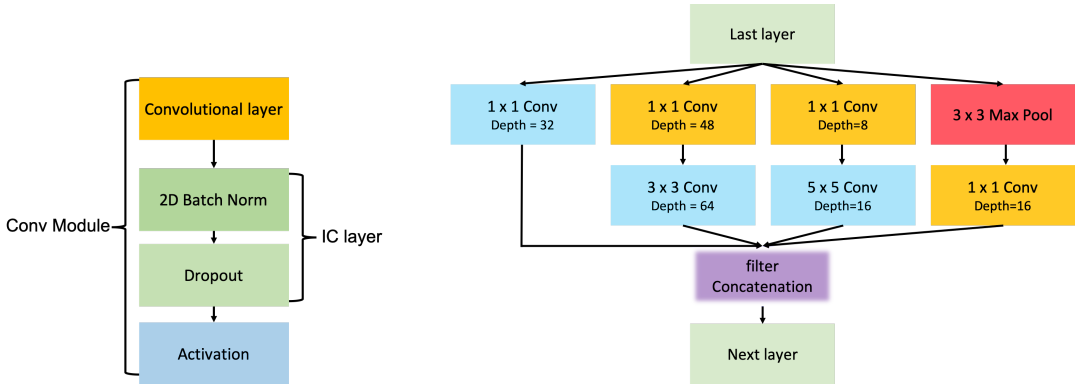


FIGURE 5.4: Left: The Conv module I used in this work, which is inspired by Li et al. (2018). The *IC layer* refers to the Independent Component layer (Chen et al., 2019); Right: The Inception module with dimension reduction (Szegedy et al., 2014).

5.2.1 Classical CNN

Zhu et al. (2014) were able to train their pulsar identification algorithms with 3,756 labelled 48 × 48 images samples using a slightly modified LeNet-5 CNN architecture. LeNet-5 is one of the earliest Convolutional Neural Networks, demonstrating high success in digit/character recognition tasks (Lecun et al., 1998a). The network has a simple 7-layer architecture, with 2 convolutional layers, 2 pooling layers, and 3 fully-connected layers. In this work, I start with a modified version of LeNet-5, including one extra convolutional layer, see Table 5.3. This extra convolutional layer is followed by a downsampling that differs between the FIRST and NVSS survey images: FIRST images are downsampled to 25 × 25 = 625, while NVSS images are downsampled to 9 × 9 = 81. Rather than using the Mean Squared Error (MSE) originally proposed for LeNet-5, I use the now more common cross-entropy loss function to train our logistic regression algorithms. The layers shown in Table 5.3 form the base architecture of all networks used in this work and I will refer to specific layer numbers from Table 5.3 whenever I manipulate or replace any functionality in the following sections.

Although such a network is sufficient to train a model, previous deep learning attempts at classifying radio galaxy morphology have applied additional regularization methods in order to improve their model generalization (test) error and avoid model over-fitting (Goodfellow et al., 2016b). In this work, I apply the independent component (IC) layer regularization strategy of Li et al. (2018) to all convolutional layers in our network and this is described in more detail in the following section. In addition, I include a dropout layer before each fully-connected layer, with the exception of the output layer.

5.2.2 Independent Component Layer

As deep learning develops, one important issue is how to train complex networks with higher efficiency (Ioffe & Szegedy, 2015; Chen et al., 2019). Among all the techniques available, Batch Normalization (BN; Ioffe & Szegedy, 2015) and Dropout (Srivastava et al., 2014) are perhaps most frequently used by radio galaxy related deep learning approaches (e.g. Aniyan & Thorat, 2017; Ma et al., 2019).

Batch normalisation is able to normalize the net activations and have their mean and unit variance become zero (Chen et al., 2019). The purpose of applying such an approach is to reduce the internal covariate shift, in other words the change in the distribution of network activations due to the change in the network parameters during training (Ioffe & Szegedy, 2015). The technique therefore is able to speed up network training, regularize model performance, and further induce a stable predictable behaviour during gradient descent (Santurkar et al., 2018).

Dropout, on the other hand, performs regularization in a different way. It introduces random gates for all inputs to a given layer, where each neuron has a probability, p, to be set to zero. Such a measure is able to remove weakly connected neurons, and has been demonstrated to regularize network performance and prevent neuron co-adaptation (Srivastava et al., 2014).

The Independent Component layer (IC) is a recently developed technique incorporating both of these techniques that has been proposed to boost model training efficiency and improve model stability (Li et al., 2018; Chen et al., 2019). Each IC layer contains a stacked combination of BN and Dropout layers, see Figure 5.4, which has been proven to be able to reduce the mutual information and the degree of correlation between any pair of neurons. Such techniques achieve more stable training behaviour and IC networks typically have their generalization ability improved (Chen et al., 2019). A recent approach put IC layers before the activation layers (Li et al., 2018), and found that such method could boost model performance comparing with those only insert a BN layer between an convoluational and an activation function.

Inspired by Li et al. (2018), in this work I replace each convolutional layer with the combination of a convolutional layer, an IC layer and an activation function and refer to this as a *Conv module*. This is illustrated in Figure 5.4. I did not use the Chen et al. (2019) strategy to maintain image input completeness.

5.2.3 Inception Module

The *Inception module* was created by Szegedy et al. (2014) and is named following a previous approach referred to as Network in Network (NIN; Lin et al., 2013). Unlike classical CNN approaches that have all of their convolutional layers stacked sequentially, the Inception module has 4 'branches': a convolutional layer with limited kernel sizes of 1×1, 3×3, 5×5 and a 3×3 max pooling layer. The 4 branches in each module operate in parallel on the same input feature map. Outputs from each 'branch' then are concatenated together and serve as the input to the next layer, see Figure 5.4.

In order to give the network improved representation power, Szegedy et al. (2014) further introduced a 1×1 convolutional layer both before the 3×3 and 5×5 convolutional layers, and after the max pooling layer. An Inception module with these additional layers is known as an *Inception module with dimension reduction* (Szegedy et al., 2014). The addition of these 1×1 convolutional layers can reduce the number of input channels, hence lowering the computational complexity by serving as dimension reduction module, whilst increasing network depth at the same time. GoogLeNet, the first architecture equipped with the Inception module with dimension reduction, won the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14; Russakovsky et al., 2014), and its use has been attempted in astronomical studies such as supernovae classification (Brunel et al., 2019) and Faraday spectra classification (Brown et al., 2019), the mapping between simulation based galaxy cluster distributions and the underlying dark matter distribution (Zhang et al., 2019a), and classification of Faraday depth complexity (Brown et al., 2018). In this work, I explore its potential when creating Architecture G, see Table 5.4 and Section 5.2.5.

5.2.4 Multi-domain CNNs

Sources with class NOM and GRG are clearly separated in physical linear size, see Figure 5.1. When looking at sample host galaxy redshift and their LAS distributions, however, the class separation becomes less distinct, see Figure 5.2. It can be seen that host galaxy redshift distributions of the two classes are significantly overlapped. Sample source LAS are more separated: Only 1.9% and 1.1% of samples in the GRGNOM-A and GRGNOM-B respectively have their source LAS between 50 and 75 arcsec. If a network is able to recognize radio components of the target source and estimate its LAS from the image data, it should be able to identify sample class of GRGNOM-A/B reasonably well. However, a source might exhibit different emission structure between its NVSS and FIRST images due to structure appearing on a range of scales, and classifier performance might differ according to that. It is also unclear how the involvement of host galaxy redshift would affect model performance.

To account for these possibilities, I modify the original network architecture, allowing multiple inputs to train together. Architectures with multiple inputs are also known as *multi-domain* neural networks, and were originally proposed by Amerini et al. (2017). They introduced both spatial domain data (2 dimensional) and frequency domain data (1 dimensional) as network inputs, with each domain training on an independent branch with their outputs concatenated into a single fully-connected layer. For such a multi-domain network, model inputs can be 2D images, 1D arrays, or scalars. Networks with multiple inputs are sometimes considered to be a variant of multi-branch networks, and I discuss this further in Section 5.2.5.

Including source redshift

Source redshift, z, as a numerical parameter, is one of the two key parameters used when identifying GRGs, as it determines the distance to the radio galaxy and hence the conversion of projected angular size to projected physical size. It is therefore intuitive to include source redshift into our models. However, given that numerical data cannot be passed through layers expecting 2-dimensional inputs, I specify for this parameter to join the training at Layer 7, see Table 5.3. Specifically, I concatenate the down-sampled outputs from Layer 6 of the network with the source redshift, an example of which is shown in Figure 5.5. Source redshifts are normalized to lie in the range 0 to 1, consistent with the normalisation of image feature data. This normalisation is discussed further in Section 5.3.2.

Multiple Image inputs

In addition to combining image data with numerical features such as redshift, I also expand the multi-domain approach further to include additional image inputs. Using this approach, images of radio galaxies observed by multiple surveys with different angular resolutions are able to be learned together. Such an approach has previously been considered with in the field of neuroscience (Aslani et al., 2018).

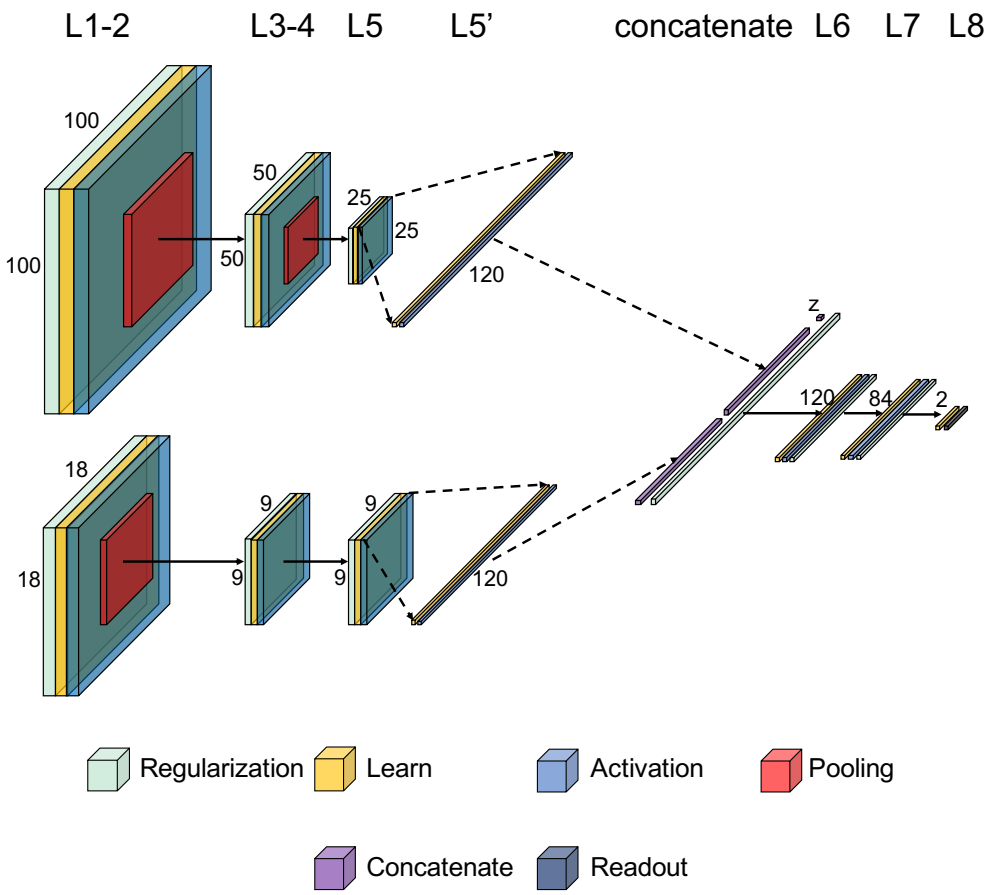


FIGURE 5.5: The network illustration of the architecture **F** in our work. In this diagram, `Regularization` refers to the use of IC layer for the convolutional layers (L1,L3 and L5), and dropout layer for fully-connected layers. `Learn` layers include convolutiona layers (L1, L3 and L5) and fully-connected layer (L6-8). `Activation` layers are all ReLU, while `Pooling` layers in the diagram are max-pooling layers. `Concatenate` operation implies that outputs from the last layer and the extra imported parameter (host galaxy redshift) would be concatenated as an 1-D vector input for the next fully-connected layer (L6). Finally, the `Readout` layer is where softmax fucntion is operated, which provides the model class probability prediction.

For full parametric details of this architecture, see Table 5.3 and Table 5.4.

In order to implement this strategy, I combine elements of the two CNNs described in the previous section together, see Figure 5.5. The depth of linear neuron concatenation remains the same as in the *Image* + z methods. By using the resulting architecture, both NVSS and FIRST images of a single source will have 120 features extracted from each survey that are then concatenated and passed into the final two fully-connected layers. In this scenario I add an additional fully-connected layer with size 120 after Layer 6, in order to increase the learning ability of the network given the larger volume of input data as well as to regularize the algorithm further.

With this architecture it is trivial to also include source redshift, z, see Figure 5.5. By adding the source redshift in the same way as in other network architectures, Layer 6 of the network has 241 neurons, while other layers remain unchanged.

5.2.5 Multi-branched CNN

The model architectures I have described so far are top-down architectures, where the input to each layer comes from the output of the previous layer. Such architectures require their convolutional layers to have a customized kernel and stride size, which can be restrictive when one wants to simultaneously learn general features with different kernel sizes. It is constraints such as these that motivated the invention of CNNs with branched modules and later Multi-branch CNNs (e.g. Li et al., 2017; Georgakilas et al., 2020).

The very early and well-known branched CNNs include the GoogLeNet and later Inception networks (e.g. Lin et al., 2013; Szegedy et al., 2014). These networks implemented the Inception module structure described in Section 5.2.3. Although the Inception module can be beneficial in terms of model performance, such an architecture can lead to large scale outputs, requiring heavy computation power.

In this work, in order to minimise training costs, I adopt the modified Inception Module and use it to replace the final convolutional layers in Architecture F, denoted L5 in Figure 5.5, and thus enable those layers to learn features with diverse kernel sizes. The filter dimensions for each layer of the Inception module in this work are chosen to be half of the equivalent value for the 'inception (3a)' model of GoogLeNet (Szegedy et al., 2014), making the output parameter number comparable to that of Layer 6 in Architecture F.

5.3 Discussion

5.3.1 Model Evaluation Metrics

In the context of deep learning classification algorithms, popular model evaluation metrics include Accuracy, Recall, Precision, F_1 score and AUC score, see e.g. definitions in Appendix A of Bowles et al. (2021). These metrics have been widely used previously in the literature to evaluate CNN based radio galaxy classifiers (e.g. Aniyan & Thorat, 2017; Ma et al., 2019; Tang et al., 2019; Bowles et al., 2021).

Performance metrics for the data set as a whole (Accuracy, AUC) and class-specific performance metrics for the GRG class (Precision, Recall) for each of the models considered in this work evaluated against the GRGNOM-A data set are listed in Table 5.5 and against the GRGNOM-B data set in Table 5.6.

5.3.2 Model Performance

Models trained with GRGNOM-A

Given that the GRGNOM-A data set has a severe class imbalance, class predictions are expected to be biased in the early phases of model training. This can be seen in Figure 5.6, where the models used in this work tend to predict almost all validation samples as class NOM in the first 200-400 training epochs, although the models do gradually overcome this issue as the training continues.

Looking at the model loss curves that used the testing set as validation set, it can be seen from Figure 5.6 that models trained with architectures A and C have their validation loss saturated quickly, while their ability to classify GRG objects increases gradually as training continues. On the other hand, the NOM recall for these models drops from 100% to around 98% and then becomes stable. In other words, the mild improvement in performance seen from architectures A and C in terms of validation accuracy is partly contributed by the improvement of GRG recall but at the expense of NOM recall.

I also note that the architectures that use only NVSS images as their input tend to exhibit more stable training and result in a higher rate of correctly classified GRG samples. This can be seen in both Figure 5.6 and Table 5.5. Compared to Architectures B and D, which have only FIRST data as an image input, models trained with only NVSS data as image inputs (Architectures A and C) have higher GRG recall by 20 to 26 % on average. This is likely to be caused by the GRG sample selection of GRGNOM-A, as shown in Figure 5.8, where GRG objects are cross-validated using NVSS images, and thus their radio components are more clearly visible compared to those of their FIRST image counterparts.

The inclusion of host galaxy redshift as an input feature has boosted model performance regardless of architecture. For architectures that with or without redshift (A & C, B & D, E & F), it can be seen that the inclusion of redshift information causes a marginal improvement in model accuracy, AUC value and GRG class metrics (as seen in Table 5.5).

Although the selection of image inputs gives different performances when using single image input models, using both image inputs generally contributes to better classification results when they are imported together. Both with and without host galaxy redshift information or the presence of Inception modules, I found that architectures E, F and G outperform the single image input models across model AUC values and GRG class precision, along with similar or better model test accuracies comparing with those single image input models. This is consistent with what has been found from other non-astronomical applications, that multi-branched approaches can boost model performance compared to the classical single input approach (e.g., Li et al., 2017; Georgakilas

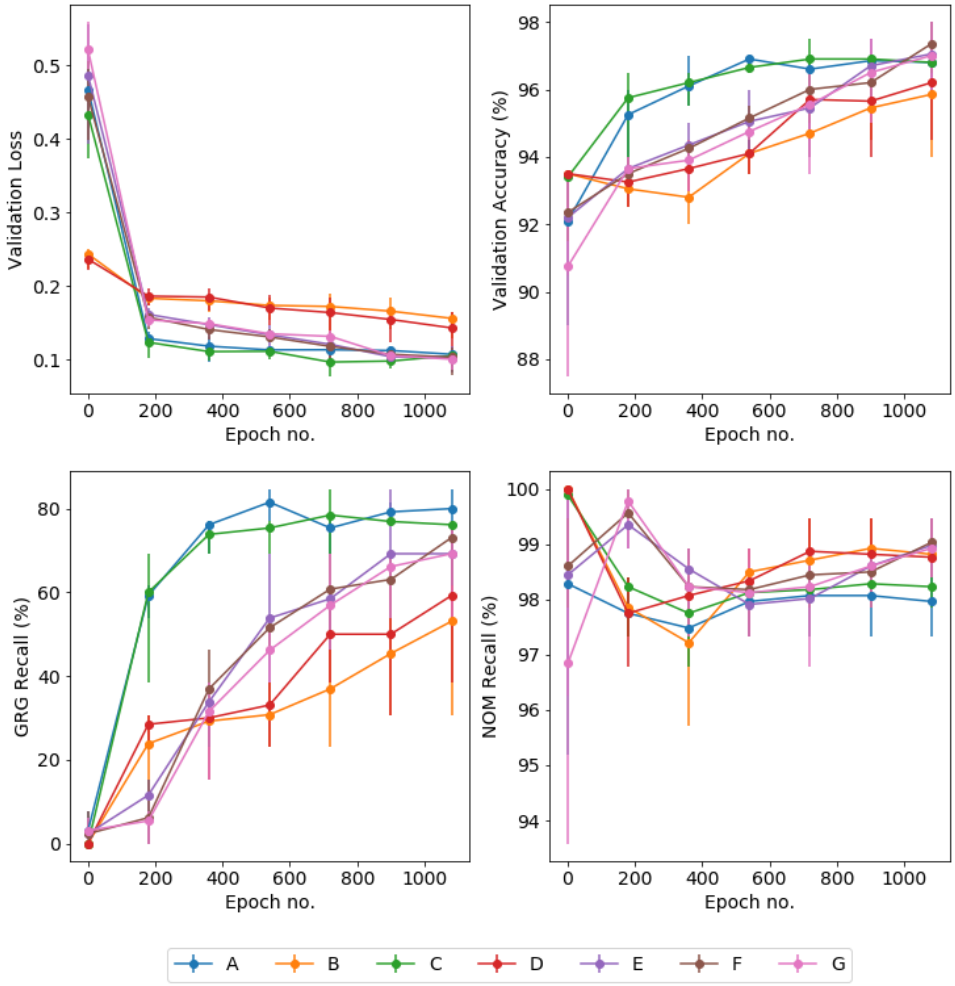


FIGURE 5.6: The averaged learning curve of the model architectures trained and validated with GRGNOM-A. Assuming normal distribution, the asymmetric errors of each data point on the diagram has covered 60% of data distribution.

et al., 2020). However, it is also noteworthy that the inclusion of multiple image inputs have on average decreased GRG class recall by 8%, implying that such multi-branch approach should be treated in caution as the classifier ideally is to identify as many GRGs as possible.

When I introduce the Inception module with dimension reduction to the model, I do not find a significant improvement in model performance when testing against the GRGNOM-A test sample. However, I find that this architecture performs differently when trained and test on GRGNOM-B, which I will discuss in Section 5.3.2.

Models trained with GRGNOM-B

By comparing Table 5.5 and Table 5.6, it can be seen that models trained using the GRGNOM-B data set are able to make more stable predictions. The involvement of Dabhade et al.

Architecture	A	B	C	D	E	F	G
Input data	NVSS	FIRST	NVSS & z	FIRST & z	NVSS & FIRST	NVSS & FIRST & z	NVSS & FIRST & z
Accuracy (%)	96.8 ± 0.5	95.7 ± 1.4	97.2 ± 0.6	96.4 ± 1.2	97.2 ± 0.8	97.2 ± 0.8	97.0 ± 0.9
AUC	0.950 ± 0.017	0.938 ± 0.025	0.956 ± 0.016	0.954 ± 0.024	0.970 ± 0.014	0.973 ± 0.015	0.968 ± 0.016
Precision (GRG)	0.739 ± 0.044	0.735 ± 0.133	0.783 ± 0.054	0.794 ± 0.109	0.832 ± 0.070	0.834 ± 0.068	0.805 ± 0.074
Recall (GRG)	0.796 ± 0.051	0.528 ± 0.164	0.791 ± 0.064	0.588 ± 0.151	0.714 ± 0.094	0.718 ± 0.098	0.708 ± 0.1
F1 score (GRG)	0.765 ± 0.036	0.608 ± 0.15	0.785 ± 0.046	0.670 ± 0.133	0.766 ± 0.073	0.768 ± 0.076	0.750 ± 0.077

TABLE 5.5: Summary of model performance metrics for all architectures trained and tested with the GRGNOM-A dataset.

Architecture	A	B	C	D	E	F	G
Input data	NVSS	FIRST	NVSS & z	FIRST & z	NVSS & FIRST	NVSS & FIRST & z	NVSS & FIRST & z
Accuracy	84.5 ± 1.3	90.0 ± 1.1	85.4 ± 1.3	89.8 ± 1.2	88.7 ± 1.2	89.1 ± 1.3	89.1 ± 1.2
AUC	0.872 ± 0.015	0.930 ± 0.010	0.890 ± 0.015	0.927 ± 0.010	0.925 ± 0.009	0.927 ± 0.011	0.929 ± 0.011
Precision (GRG)	0.763 ± 0.039	0.831 ± 0.038	0.784 ± 0.039	0.825 ± 0.035	0.791 ± 0.032	0.808 ± 0.032	0.801 ± 0.032
Recall (GRG)	0.569 ± 0.036	0.766 ± 0.035	0.592 ± 0.033	0.766 ± 0.034	0.761 ± 0.033	0.752 ± 0.041	0.761 ± 0.037
F1 score (GRG)	0.651 ± 0.031	0.796 ± 0.022	0.674 ± 0.029	0.794 ± 0.023	0.775 ± 0.024	0.778 ± 0.029	0.780 ± 0.026

TABLE 5.6: Summary of model performance metrics for all architectures trained and tested with the GRGNOM-B dataset.

(2020b) sample data in the training set lowers the class imbalance ratio from 14:1 in GRGNOM-A to around 3:1 in this data set. With more GRG examples in the training set, models are able to learn more quickly and make more stable predictions, see Figure 5.7.

The biggest difference between Figure 5.6 and Figure 5.7 is the reversal of model performance differences between single image input models trained with NVSS images and FIRST images. Compared with Architectures A and C, Architectures B and D have lower test losses and higher test accuracies after 180 epochs of training. The largest contribution to this difference can be attributed to data sample selection. The 101 Dabhade et al. (2020b) samples in the GRGNOM-B training set are identified from LoTSS survey maps with an angular resolution of $6''$, and consequently the source morphology of these objects is found to be much closer to that of the FIRST survey with an angular resolution of $5.4''$, rather than the lower resolution NVSS survey. Such similarity in angular resolution will contribute to model performance: once FIRST image inputs are imported, models trained with GRGNOM-B data samples receive F1 scores higher than 0.76 on average, see Table 5.6. Moreover, models trained using only FIRST images as inputs have GRG Recall/Precision values $\geq$19.7/6.8% higher than those trained with equivalent NVSS images (A & B).

The inclusion of host galaxy redshift too have generally provided similar (B & D) or better (A & C, E & F) model performance when trained and tested using the GRGNOM-B data set. The influence of the multi-branch network approach, however, appeared to behave differently. Architecture E was found to have model performance in between Architecture A and B.

Interestingly, the inclusion of the Inception modules also seems to mildly improve model performance when testing with the GRGNOM-B data set. This is perhaps also due to the higher resolution sample selection for this data set. The extra network parameters are able to learn more complex source morphology features from these samples, and thus have slightly boosted model performance relative to other architectures.

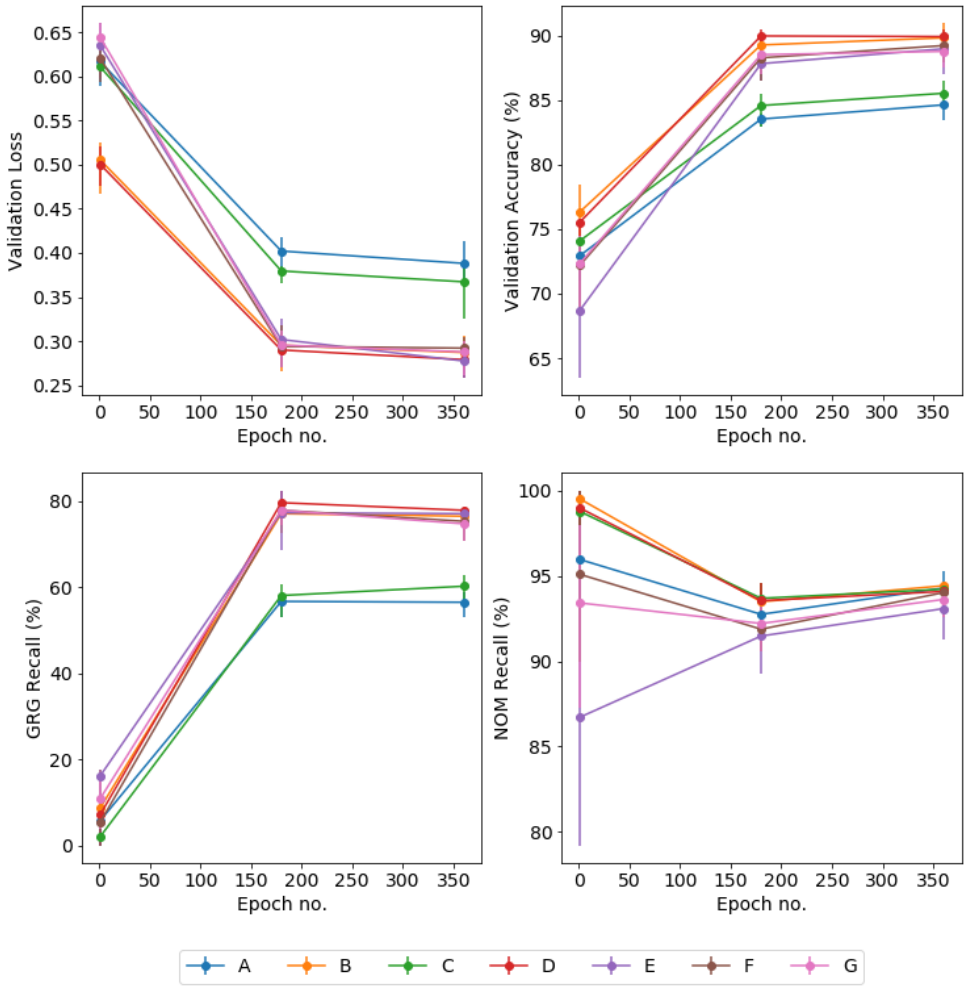


FIGURE 5.7: The averaged learning curve of the model architectures trained and validated with GRGNOM-B. Assuming normal distribution, the asymmetric errors of each data point on the diagram has covered 60% of data distribution.

Architecture	A	B	C	D	E	F	G
Input data	NVSS	FIRST	NVSS & z	FIRST & z	NVSS & FIRST	NVSS & FIRST & z	NVSS & FIRST & z
Accuracy	81.0 ± 0.8	83.6 ± 1.4	81.0 ± 0.8	83.6 ± 1.3	81.6 ± 1.0	81.8 ± 0.9	81.6 ± 1.0
AUC	0.790 ± 0.018	0.830 ± 0.019	0.800 ± 0.021	0.832 ± 0.019	0.808 ± 0.018	0.806 ± 0.021	0.805 ± 0.021
Precision (GRG)	0.862 ± 0.045	0.874 ± 0.039	0.873 ± 0.044	0.886 ± 0.046	0.876 ± 0.050	0.895 ± 0.042	0.873 ± 0.049
Recall (GRG)	0.306 ± 0.024	0.416 ± 0.052	0.300 ± 0.03	0.411 ± 0.046	0.327 ± 0.033	0.325 ± 0.030	0.328 ± 0.033
F1 score (GRG)	0.451 ± 0.028	0.562 ± 0.05	0.445 ± 0.034	0.560 ± 0.047	0.475 ± 0.036	0.476 ± 0.035	0.476 ± 0.038

TABLE 5.7: Summary of model performance metrics for all architectures trained with the GRGNOM-A dataset and tested with the model generalization test set described in Section 5.3.2.

Generalization Ability

To this point I have evaluated the models in this work using test data taken from the same underlying data set as the training data, either GRGNOM-A or GRGNOM-B. To evaluate model generalization ability more broadly I now consider models trained using GRGNOM-A and tested using data from the 51 GRG samples in the GRGNOM-B test set. I also test these models with another 149 RGZ DR1 samples of class NOM that are not found in either training set. I refer to the resulting test set as GRGNOM-Gen. I do not consider the opposite approach as the GRG samples in the GRGNOM-A test set have been included in the GRGNOM-B training set. These evaluation metrics can be seen in Table 5.7.

From Table 5.7 it can be seen that when making predictions on these test samples, models trained with GRGNOM-A perform comparatively less well in terms of general model metrics than those directly trained on the GRGNOM-B data set. A similar situation also occurs when looking at GRG recall and F_1 score; even the best performing Architecture B can only provide a GRG recall of 0.416 ± 0.052. On the other hand, the same architecture has a GRG precision of 87.4% on average, implying that the model is able to identify NOM class objects well. In other words, although these models achieve higher GRG classification precision when compared to those trained with the GRGNOM-B training set, they are unable to reach a comparably high classification completeness. In order to find the majority of the GRGs in the Dabhade et al. (2020b) sample, it is still essential to have some of the Dabhade et al. (2020b) samples included in model training set.

In our previous discussion of different architectures, I noted that including redshift information has provided similar or better improvements in model performance, depending on architectures. However, I did not observe comprehensive improvement when testing on GRGNOM-Gen: models trained with redshift have gained 1-2% improvement in GRG Precision, with the expense of 0.2-0.6% decrease in GRG Recall. The advantage when including multi-domain data did not hold when testing on GRGNOM-Gen, resulting in a 0.2% improvement in GRG precision, 0.002 for model AUC value with the expense of 2% in model testing accuracy and 0.9% decrease in GRG recall (e.g., B & E). Finally, when comparing Architectures F and G, the inclusion of inception modules behaves slightly better (1% improved in GRG recall) when testing on the GRGNOM-Gen.

Angular Size Distance vs. Host galaxy redshift

A consideration when introducing host galaxy redshift as an input feature is that the relationship between host galaxy redshift and angular size distance, D_A, is not strictly linear. As an experiment, I used the equivalent D_A in Gpc to replace host galaxy redshift when training Architecture F using the GRGNOM-A data set. The resulting models have an average AUC of 0.973 ± 0.015, slightly lower than but not significantly different from that found when using z directly.

When looking at GRG classification performance, the D_A alternative returns a GRG Precision of 0.830 ± 0.063 and a GRG Recall of 0.748 ± 0.097. Comparing these metrics with the architecture F in Table 5.5, which use z directly, it can be seen that the GRG precision was improved by 3.2% with the expense of 13.2% decrease of GRG recall. This suggests that the network architecture already has sufficient capacity in its trainable parameters to learn the redshift - D_A relationship, or an approximation of it.

5.3.3 Common features shared by the misidentified samples

The model evaluation I have presented so far is based on a simple assumption: that the GRGNOM-A/B data sets are fully understood, reliable and confidently labelled. Yet it is unclear whether frequently misclassified objects in our models share common features. In this work I train each model architecture 10 times on the GRGNOM-A/B training sets using independent Xavier initializations. By applying each of these models individually to all samples in the GRGNOM-B test set I am able to identify all GRGNOM-B test samples that have a misclassification rate of $\geq$50% for any architecture used in this work. These samples are summarised in Table 5.8.

A potential data 'trap' in the GRGNOM-B data sets comes from the Dabhade et al. (2020b) sample. These objects were identified using the 151 MHz LoTSS survey (Shimwell et al., 2019). Considering that $S \propto \nu^{\alpha}$, where $\alpha = -0.7$ for optically thin synchrotron emission, sources will be brighter at 151 MHz compared to their NVSS or FIRST counterparts at 1.4 GHz. In addition, the median rms of LoTSS is 71 μJy beam^{-1}, lower than half that of the FIRST survey and around 16% of the NVSS sensitivity of 0.45 mJy beam^{-1}. This means that LoTSS will be more sensitive to faint radio emission, and that some radio structures present in LoTSS images might be missed in the equivalent NVSS and/or FIRST images. Besides the 'trap', the aforementioned image specifications, pre-processing choices, selection of input domains (NVSS images, FIRST images, host galaxy redshifts), and selection of architectures could also result in differences in model performance.

In order to investigate these frequent misclassifications I use the data traceability built into our data set, see Section 5.1.1. Similarly traceable data sets have been built and implemented for a number of recent deep learning studies (e.g. Wu et al., 2019; Walmsley et al., 2020) and their data traceability used to explain why some samples are mistakenly identified in a frequent manner (e.g. Wu et al., 2019). In this case, the sources listed in Table 5.8 were traced using their unique object IDs, see Section 5.1.4. I then analysed

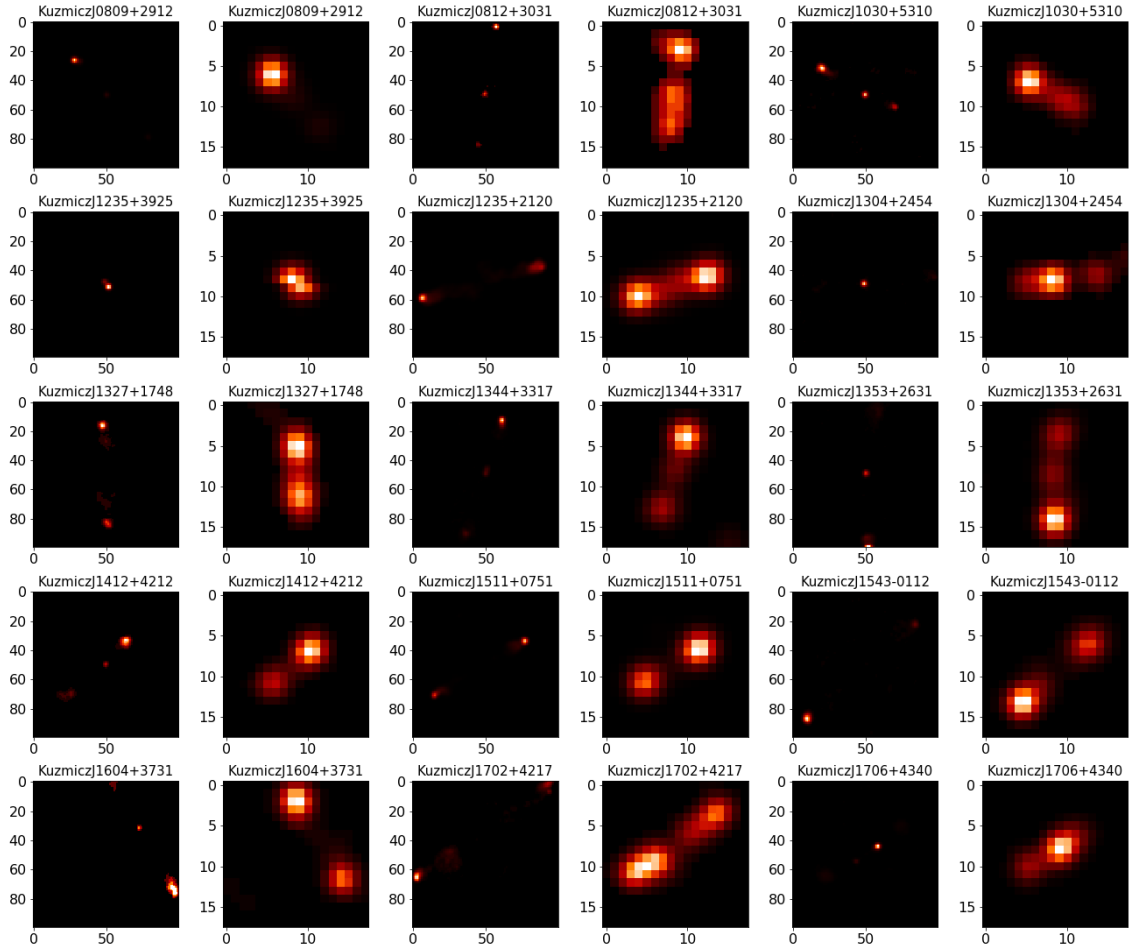


FIGURE 5.8: Example **GRG** images extracted from the Kuźmicz et al. (2018a) catalogue. Under the same object ID, the left image refers to its pre-processed FIRST image, while the right image is the pre-processed NVSS image of the object.

these cases of object misclassification by looking at their NVSS and FIRST pre-processed images as well as the host galaxy redshift in each case.

Low surface brightness

As described in Section 5.1.3, the sigma-clipping performed during image pre-processing will replace all pixels with values lower than the local 3-$\sigma_{\rm rms}$ level with zeros, resulting in faint objects that have only a mild luminosity difference with the noise background becoming even fainter in the normalized image when a secondary source is present in the field of view. For example, the source Dabhade 237 was faint but visible at 151 MHz in the original LOFAR data (FigureA.8; Dabhade et al., 2020b). However, the both the radio core and the nearby radio lobes seem to be too faint to be clearly identified in the pre-processed FIRST and NVSS maps used in this work. The models are therefore unable to find a GRG like object in the image, but instead identify a secondary source at the edge of the field to be class NOM.

Object ID	≥50% Mistakenly Identified Architectures
Dabhade230	All
Dabhade217	All
Dabhade198	All
Dabhade173	All
Dabhade237	All
Dabhade186	All
Dabhade216	All
Dabhade193	All
RGZJ080417.6+320250	A,B,C,E,F,G
RGZJ075855.6+360246	A,B,C,D,E,F
Dabhade204	A,C,E,F,G
RGZJ075030.7+525022	A,C,E,F,G
RGZJ080448.0+081254	A,C,E,F,G
RGZJ075539.6+160158	A,C,E,F,G
RGZJ075157.7+212049	B,D,E,F,G
RGZJ080427.8+132930	A,D,E,F
RGZJ075812.7+190043	B,D,E,G
Dabhade185	A,B,C,D
Dabhade221	B,D,F
Dabhade201	A,B,D
Dabhade197	A,C,G
Dabhade214	B,F,G
Dabhade199	A,C
Dabhade227	A,C
Dabhade206	A,C
Dabhade226	A,C
RGZJ074627.1+174337	A,C
RGZJ074720.7+335008	A,C
Dabhade229	A,C
Dabhade231	A,C
Dabhade220	A,C
Dabhade209	A,C
Dabhade163	A,C
RGZJ080404.5+153334	B,D
RGZJ075306.1+121504	B,D
Dabhade210	E,G
RGZJ080402.5+452258	E
RGZJ075620.0+301630	E

TABLE 5.8: A summary of frequent mistakenly identified GRGNOM-B testing samples in this work. A sample would be included in this table if it has over 50% rate to be mistakenly identified in at least one architecture adopted in this work.

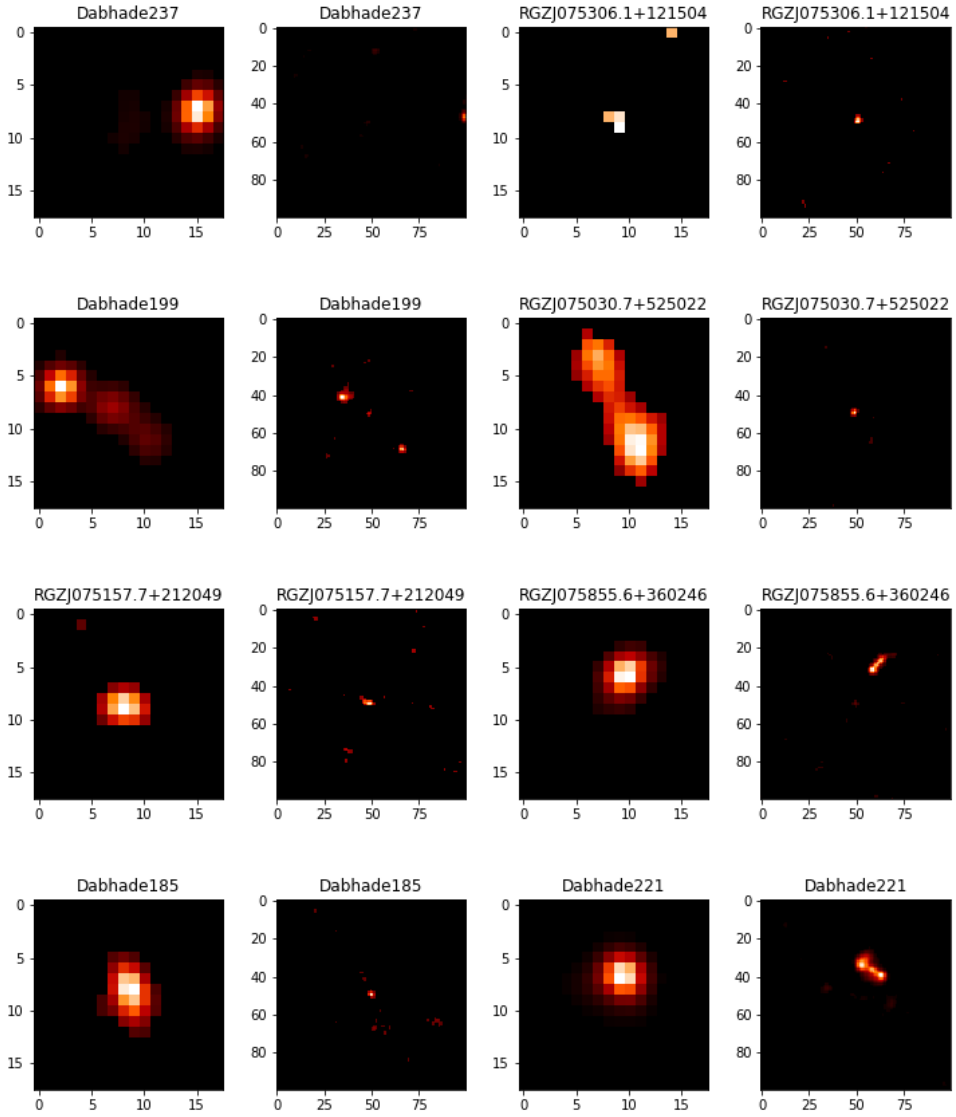


FIGURE 5.9: A summary of misidentified GRGNOM-B testing samples, which represent the typical types of misclassification described in Section 5.3.3.

Input Domain Selection

In general, multi-domain approaches are expected to be able to help a network to cross validate its prediction results. In this work, 39% of the frequently misidentified objects listed in Table 5.8 have their identification corrected when both image domains are imported. The sources RGZ J075306.1+121504 and Dabhade 199 are two representative examples: RGZ J075306.1+121504 is frequently misidentified without the inclusion of its NVSS image. In spite of the slightly extended structure along with scattered faint background emissions in its FIRST image, its NVSS image retains clear compact structure, helping the networks to correctly identify the object class. On the other hand, Dabhade 199, whilst still having segments of its lobes resolved out in the FIRST image, has a clearly extended central source. In this case, the lower resolution NVSS image has mistakenly included two objects: Dabhade 199 as a GRG and another compact source, which makes it difficult for the network to identify it as a GRG with confidence; however, with the help of the additional information from the FIRST image, the networks are able to classify this source correctly in most cases.

Conversely, the use of multiple image domains may also have a negative impact on model performance. Two examples of this are RGZ J075030.7+525022 and RGZ J075157.7+212049. While RGZ J075030.7+525022 looks compact in its FIRST image, the object shows a well-extended morphology in its NVSS map. Such difference in sample morphology has made any architecture considered NVSS image input unable to make correct classification. On the other hand, RGZ J075157.7+212049 has shown extended morphology along with significant scattered emissions in its FIRST map, while its NVSS map is dominated by a single compact object. Architectures considered sample FIRST image data then would make incorrect identifications.

Architecture bias

As well as the selection of input data, differences in network architecture will also contribute to model performance. RGZ J075855.6+360246 ($z = 0.336$; see Figure 5.9) was correctly identified only when the Inception module was introduced. Both of its NVSS and FIRST images have included a nearby well-extended object, while itself only appears in its FIRST image, looks compact and faint. Considering the uniform kernel size of 5 for the convolutional layers in the models of architectures A to F, these models might have found it difficult to capture the morphological features of the small and faint object. However, by introducing an Inception module to network of architecutre G, the network was able to capture features down to 1-pixel scale. This perhaps explain why the object got correctly identified only when I use architecture G.

5.3.4 Cases requiring further explanation

Besides the aforementioned examples, there are frequently misclassified objects in Table 5.8 that I found difficult to explain by simply looking at object sample data. For

instance, Dabhade 185 can be seen as a unique example of multi-domain cross validation. Its radio lobes in the FIRST image for this object has been resolved out, and become unresolved in the NVSS image map. This object was classified as GRG when both NVSS or FIRST was used, which I find it difficult to interpret.

Another issue I met when trying to explain model behavior happens when there is more than one well-extended object in an image map. From Table 5.8 it can be seen that the source Dabhade 221 was consistently misclassified by three of the models in this work. In this case, via visual inspection I found that aside from the GRG, there is another well-extended radio source visible on its FIRST image, see Figure 5.9. The translational equivariance of the convolutional layers does not require a source to be located at the centre of the image in order to be correctly classified and this is an example of such a circumstance. Which source emission finally contributes to the prediction remains uncertain.

Analysis of such complex cases could perhaps benefit from the use of explainable Artificial Intelligence (XAI) tools (e.g. SHAP; Lundberg & Lee, 2017). Rather than visualizing feature maps for each specific layer, these tools allow users to directly visualize which features the network as a whole has recognized from each sample image, and in some cases evaluate their contributions to target class identification. Another possibility might be to use attention-gating, which produces attention maps that facilitate the interpretation of a classification choice as made by the model (e.g. Bowles et al., 2021).

5.3.5 Comparison to other automated search methods

As described in Section 1.3.1, a previous approach to automated GRG identification was made by Proctor (2016) using a decision tree based machine learning approach. Using source pair separations from the NVSS catalogue, Proctor (2016) produced a list of GRG candidates with $LAS \geq 240''$. From Figure 5.2 it can be seen that all of the GRG class objects used in this work are smaller than the limit of Proctor (2016); however, it can also be seen that there is still a clear separation in LAS between NOM class galaxies and GRGs and I suggest that it is for this reason that the inclusion of redshift information did not strongly affect the results of the models presented in this work. This LAS separation is likely to be a consequence of the historic sample selection governing current catalogues of known GRGs. The development of labelled training sets with larger numbers of intermediate size radio galaxies will enable future studies to investigate this cross-over region of parameter space in more detail.

5.4 Conclusions

Previous GRG searches have largely depended on visual inspection, and, while successful given the size of historic observational data bases, this human-powered methodology is unlikely to scale well to the new generation of radio sky survey data, which will require the investigation of millions of extended radio sources.

In this work, I have explored the possibility of automated GRG identification through deep-learning by using 7 different CNN-based model architectures, including a multi-branched CNN algorithm that incorporates information from multiple surveys with different resolution. The best performing models in this study achieve 97.2% and 90.0% test accuracy using the GRGNOM-A and GRGNOM-B test samples described in this work.

A key result from this work is the introduction of multi-domain networks in order to boost model performance compared with the classical CNN architectures that use a single type of image input. By including host galaxy redshift information, model performance was found to be improved. Importing both NVSS and FIRST images at the same time, on the other hand, got 39% of objects that were misclassified by a single domain network corrected. Finally, the use of an inception module could affect model performance under certain circumstances, and was found to be able to correct those misclassifications when the target object is small and compact.

Finally, I investigated the cause of frequent misclassifications by inspecting individual samples, and found that other than the aforementioned factors, a sample might be misclassified if (a) its sample image contains multiple sources, (b) the standard preprocessing procedures have eliminated part of its extended morphology, or (c) its radio component was partly resolved out comparing with the survey map images used to identify the GRG (e.g. LoTSS at 151 MHz).

Chapter 6

Conclusion and Outlook

Giant Radio Galaxies (GRGs) are a group of relatively rare radio galaxies, with projected linear sizes greater than 0.7 Mpc. Given their gigantic size, they are believed to be the final point of radio galaxy evolution. Discovering these objects is important not only for testing the physics of galaxy evolution but also because GRGs are found predominantly in warm-hot intergalactic medium (WHIM). As the lobes of GRGs expand, they would interact with WHIM, and thus could serve as a probe of their local environment. In the context of low-density IGM studies, it is then unsurprising that the hunt for GRGs remains important.

GRGs have historically been poorly detected in radio surveys such as NVSS or FIRST due to a combination of poor surface brightness sensitivity, field of view constraints and a tendency to build telescopes working at GHz frequencies due to technical limitations. However, as the next generation of radio telescopes, such as ASKAP and LOFAR, with large fields of view come into play, this situation is changing. To date, astronomers have discovered over 800 GRGs, and the number is still growing.

In general, the process of identifying a GRG can be separated into 5 steps: radio component identification, host galaxy identification, radio source LAS measurement, host galaxy redshift measurement, and source projected linear size calculation. In order to investigate additional source properties, scientists will usually also perform further deep imaging observations of their target GRG candidate in the radio or optical wavebands. If necessary, they would also observe the host galaxy spectra for a candidate and have its spectroscopic redshift measured. Also, given that investigating the radio morphology of a DRAGN may benefit our understanding of the source formation mechanisms for GRGs or our understanding of the environmental influences upon such a source, scientists will typically also classify source radio morphology (i.e. using the FR classification scheme) as well.

For making both GRG candidate identifications and source radio morphology classifications, the traditional method is visual inspection. By looking at sample image maps at different observational frequencies, experts with prior knowledge of radio galaxies are able to connect radio components belonging to the same DRAGN, identify their FR

morphology (facilitated by measurements of host-hot spot relative position), and cross-validate DRAGN host galaxies with high reliability. Such methods have been widely implemented over the last few decades using radio surveys like NVSS, FIRST and SUMSS, each of which contains entries on the scale of 1-2 million sources. Under customized sample selection rules, experts are able to look through the complete sky survey image data and perform source identification.

The anticipated work load associated with this process in the era of big data is expected to change the game completely. EMU, as one of the eight ASKAP large scale radio surveys, is expected to discover 70 million radio galaxies, of which 10% will require visual inspection. In order to tackle the barriers caused by such gigantic data sample sizes, the RGZ team worked with over 12 000 project citizen scientists together to identify DRAGNs. Citizen scientists are asked to connect the radio components of the same DRAGN and to identify its infrared counterpart from $3\times3\,\mathrm{arcmin}^2$ cutouts, where each cutout has FIRST radio contours starting at $3\,\sigma_{\mathrm{rms}}$ on top of a *WISE* 3.4 μm image. Moreover, they would discuss possible GRG candidates through an online forum, `RadioTalk`. Although such practice did help scientists discover a few new GRGs, unfortunately, it has shown that crowd-sourced visual inspection itself is insufficient to identify source morphological class or perform GRG identification on reasonable timescales when faced with tens of millions of sample objects.

Given the limitations of the crowd-sourced approach, it is unsurprising that astronomers have embraced automated algorithm development. Recently, several studies have made efforts to develop machine learning based classification algorithms. As part of this work, they have also built several data sets to support their machine learning algorithm development. These algorithms are found to be able to identify radio galaxy morphological class (i.e. FR I, FR II) with human-comparable accuracy. However, these practices have been focused on single survey data. It is unclear whether the 'knowledge' learned from these algorithms could either be used directly to classify source morphology on other survey data, or to benefit algorithm development trained with other survey data (i.e. boost model performance).

In the context of GRG identification, automated classification algorithm development is in a more early phase. The first and the only published semi-automated GRG classifier was developed in 2016. This decision tree based algorithm was designed to train on pre-selected radio component features, and was able to identify two-component objects with LAS $\geq 4'$. However, without the involvement of host galaxy redshift information, most GRG candidates predicted by the algorithm were not confirmed to be GRGs until a follow-up manual identification work presented in late 2020. Given the complex morphology of GRGs, if one wants to perform automated end-to-end class prediction, the following questions must be considered:

(i) is there an accessible and traceable machine learning data set available to enable algorithm training to separate GRG candidates from radio sources of smaller physical size?

(ii) is it possible to develop an end-to-end machine learning based classification algorithm without heavy feature pre-selection?

(iii) is it possible to train an algorithm with both survey image data and host galaxy redshift considered?

In order to answer the 'knowledge transfer' question above, in Chapter 3 I have introduced a CNN-based machine learning classification algorithm (a modified version of AlexNet). I trained the architecture with single survey image data. The dataset I used comprises a sub-sample of CoNFIG and FRICAT catalogues, with NVSS and FIRST image of each sample object available. Beyond training the model from scratch, I also explored the possibility to boost model performance by applying transductive transfer learning.

My approaches achieved human-comparable classification accuracy when testing on FIRST and NVSS images. Depending on the transfer learning method used, I have demonstrated that transfer learning models can result in even higher model accuracies or save training time by up to 79%.

A key result from this work is that inheriting model weights pre-trained on higher resolution survey data (e.g. FIRST) can boost model performance when re-training with lower resolution survey images (e.g. NVSS). Nevertheless, I found that the reverse situation instead *lowered* the model performance.

After exploring the possibility of transfer learning, I started to work on building an accessible and traceable machine learning dataset for GRG classification. The dataset should have both survey image data and host galaxy redshift information available. During the sample selection stage, I chose RGZ DR1 as the data pool for DRAGNs of smaller projected linear sizes. The data sample selection work included (a) the confirmation of data availability: NVSS, FIRST image data, source host galaxy redshift, source LAS measurement and projected linear size derivation; (b) visual inspection of possibly duplicated entries; and (c) a split between GRG candidates and objects of smaller sizes. As a bonus of the work, in Chapter 4 I identified 5 new GRGs from RGZ DR1. Four out of the 5 GRGs share an FR II radio morphology, and cover the redshift range of $0.28 < z < 0.43$.

When investigating the environment of these GRGs, I associated one of the newly identified GRGs with the brightest cluster galaxy in galaxy cluster GMBCG J251.67741+36.45295 (Hao et al., 2010). In addition, I identified a further 13 previously known GRGs to be BCG candidates from literature data. This increased the number of known BCG GRGs by more than 60% at the time my discovery got published. Interestingly, I found that the local galaxy density of these sources was significantly higher than that of non-cluster GRGs. Their existence has challenged the hypothesis that GRGs are able to grow to Mpc scales only due to locally under-dense environments.

Once the sample selection was done, I was able to build the intended dataset - GRGNOM. The GRGNOM dataset has three sub datasets: GRGNOM-A with GRG/NOM class ratio of 1:14.4, GRGNOM-B with the same ratio of around 1:3 and GRGNOM-Gen with the same GRG samples as that of GRGNOM-B testing samples for model generalization

ability test only. I then explored the possibility of developing a deep learning based end-to-end GRG classifier by using 7 different CNN-based network architectures, including a multi-domain multi-branched CNN algorithm that incorporates information from multiple surveys with different resolution. The training was made using GRGNOM-A and GRGNOM-B. The best performing models in this work achieve 97.7% and 96.7% test accuracy when predicting GRGs from the GRGNOM-A and GRGNOM-B test samples. I further tested the model generalization ability of those models trained with GRGNOM-A training samples using the GRGNOM-Gen sample. Compared to those models trained with GRGNOM-B samples directly, I found models trained with the GRGNOM-A training set shared poor GRG `Recall` when testing on GRGNOM-Gen samples.

A key contribution made by this work is the introduction of multi-branched networks aiming to improve model performance compared with the classical approach of training on a single type of image input. This is perhaps the very first multi-domain multi-branch neural network application in astronomy, and 68% of the test sample objects that were mis-classified by a single domain network became correctly classified when both NVSS and FIRST image domains were used. The inclusion of host galaxy redshift, however, was found to affect model performance only modestly. I attribute this to the clear LAS separation between the two classes of objects in the GRGNOM samples, making the involvement of redshift information redundant at present. As more intermediate size radio galaxies get discovered by more sensitive surveys in the future, I suggest that this circumstance is likely to change.

Finally, thanks to the data traceability of GRGNOM, I was able to investigate the cause of frequent misclassifications by inspecting individual samples. Beyond the aforementioned factors, I found that a sample might be frequently misclassified when (a) there exist multiple sources in a single image, (b) extended source morphology is partly eliminated during the standard image pre-processing procedure, or (c) comparing with the survey map images used to identify the GRG (e.g. LoTSS at 151 MHz), radio structure of a sample is partly resolved out in either the NVSS or FIRST map.

In conclusion, I have applied a transductive transfer learning application in radio astronomy to classify the radio morphology of DRAGNs, as well as discovering 5 GRGs from RGZ DR1 data and 13 more BCG GRGs from literature data. I developed a multi-domain multi-branch CNN algorithm to identify GRGs from DRAGNs of smaller projected linear size. These applications deserve further development, not only in the case of GRG identification/source morphology classification, but also in the other fields of astronomical research. For example, applying transfer learning to survey data from next-generation telescope arrays like LOFAR, MeerKAT, ASKAP or SKA-MID. However, further investigations must be made before applying transfer learning to survey data with ultra-high resolution such as SKA1-MID.

In the future, the improvement of multi-branch CNNs that have the algorithm embedded into a more complex network able to perform source finding, detection and classification (i.e. ClaRAN) in bulk might be necessary. Since next-generation telescope arrays such as SKA will need to process data automatically before sending the data products to

users around the world, an automated pipeline to perform source finding, identification and early classification would become essential.

Additionally, a concern when using these deep learning algorithms is their weak model interpretability. Complex deep learning algorithms have long been seen as 'black boxes', where developers find it difficult to explain a model to its potential users in a user-friendly manner. Although efforts have been made to visualize source features which contribute to sample class prediction, for example attention-gating, it is yet to be examined if any tools (i.e. XAI tools) could benefit quantitative (positive or negative) evaluation of features contributed to model classification. Efforts should also be made to consult radio galaxy experts who are not familiar with machine learning, and understand which manner of model explanation they prefer.

Bibliography

Abazajian K. N., et al., 2009, ApJS, 182, 543

Ackermann S., Schawinski K., Zhang C., Weigel A. K., Turp M. D., 2018, MNRAS, 479, 415

Adorf H.-M., Meurs E. J. A., 1988, Supervised and Unsupervised Classification - The Case of IRAS Point Sources. p. 315, doi:10.1007/3-540-50135-5_86

Aguado D. S., et al., 2019, ApJs, 240, 23

Alam S., et al., 2015, ApJs, 219, 12

Albareti F. D., et al., 2017, Astrophysical Journal, Supplement, 233, 25

Alhassan W., Taylor A. R., Vaccari M., 2018, MNRAS, 480, 2085

Amerini I., Uricchio T., Ballan L., Caldelli R., 2017, arXiv e-prints, p. arXiv:1706.01788

Amirkhanyan V. R., 2016, Astrophysical Bulletin, 71, 384

Amirkhanyan V. R., Afanasiev V. L., Moiseev A. V., 2015, Astrophysical Bulletin, 70, 45

Andreasyan R. R., Hovhannisyan M. A., Paronyan G. M., Abrahamyan H. V., 2013, Astrophysics, 56, 382

Angel J. R. P., Wizinowich P., Lloyd-Hart M., Sandler D., 1990, Nature, 348, 221

Aniyan A. K., Thorat K., 2017, ApJS, 230, 20

Aslani S., Dayan M., Storelli L., Filippi M., Murino V., Rocca M. A., Sona D., 2018, arXiv e-prints, p. arXiv:1811.02942

Baade W., Minkowski R., 1954, ApJ, 119, 206

Bagchi J., et al., 2014, Astrophysical Journal, 788, 174

Bailer-Jones C. A. L., 2000, Astronomy and Astrophysics, 357, 197

Baldi R. D., Capetti A., 2008, Astronomy and Astrophysics, 489, 989

Baldi R. D., Capetti A., 2009, Astronomy and Astrophysics, 508, 603

Baldi R. D., Capetti A., Giovannini G., 2015, Astronomy and Astrophysics, 576, A38

Baldi R. D., Capetti A., Massaro F., 2018, Astronomy and Astrophysics, 609, A1

Baldi R. D., Capetti A., Giovannini G., 2019, Monthly Notices of the RAS, 482, 2294

Banerji M., et al., 2010, Monthly Notices of the RAS, 406, 342

Banfield J. K., et al., 2015, MNRAS, 453, 2326

Banfield J. K., et al., 2016, Monthly Notices of the RAS, 460, 2376

Baum S. A., Heckman T., 1989, Astrophysical Journal, 336, 681

Bazell D., Aha D. W., 2001, Astrophysical Journal, 548, 219

Beasley A. J., Gordon D., Peck A. B., Petrov L., MacMillan D. S., Fomalont E. B., Ma C., 2002, ApJS, 141, 13

Becker R. H., White R. L., Helfand D. J., 1995, ApJ, 450, 559

Begelman M. C., Rees M. J., Blandford R. D., 1979, Nature, 279, 770

Benatan M., Pyzer-Knapp E. O., 2018, Practical Considerations for Probabilistic Back-propagation, http://bayesiandeeplearning.org/2018/papers/99.pdf

Bengio Y., Courville A., Vincent P., 2012, arXiv e-prints, p. arXiv:1206.5538

Bennett A. S., 1962a, MemRAS, 68, 163

Bennett A. S., 1962b, MNRAS, 125, 75

Bennett C. L., Lawrence C. R., Burke B. F., Hewitt J. N., Mahoney J., 1986, ApJs, 61, 1

Best P. N., Heckman T. M., 2012, MNRAS, 421, 1569

Best P. N., Kauffmann G., Heckman T. M., Ivezić Ž., 2005, MNRAS, 362, 9

Bicknell G. V., 1995, Astrophysical Journal, Supplement, 101, 29

Bilicki M., et al., 2016, ApJS, 225, 5

Bisong E., 2019, Google Colaboratory. Apress, Berkeley, CA, pp 59–64, doi:10.1007/978-1-4842-4470-8_7, https://doi.org/10.1007/978-1-4842-4470-8_7

Blanton M. R., et al., 2003, ApJ, 592, 819

Blundell K. M., Rawlings S., 2001, in Laing R. A., Blundell K. M., eds, Astronomical Society of the Pacific Conference Series Vol. 250, Particles and Fields in Radio Galaxies Conference. p. 363

Bock D. C. J., Large M. I., Sadler E. M., 1999, AJ, 117, 1578

Bolton J. G., 1948, Nature, 162, 141

Bolton J. G., Stanley G. J., 1948, Nature, 161, 312

Bowles M., Scaife A. M. M., Porter F., Tang H., Bastien D. J., 2021, Monthly Notices of the RAS, 501, 4579

Branson N. J. B. A., Elsmore B., Pooley G. G., Ryle M., 1972, Monthly Notices of the RAS, 156, 377

Bridle A. H., Perley R. A., 1984, ARAA, 22, 319

Bridle A. H., Davis M. M., Meloy D. A., Fomalont E. B., Strom R. G., Willis A. G., 1976, Nature, 262, 179

Brinchmann J., Charlot S., White S. D. M., Tremonti C., Kauffmann G., Heckman T., Brinkmann J., 2004, MNRAS, 351, 1151

Brown R. L., 1982, Astrophysical Journal, 262, 110

Brown S., et al., 2018, Monthly Notices of the Royal Astronomical Society, p. sty2908

Brown S., et al., 2019, Monthly Notices of the RAS, 483, 964

Brunel A., Pasquet J., Pasquet J., Rodriguez N., Comby F., Fouchez D., Chaumont M., 2019, arXiv e-prints, p. arXiv:1901.00461

Burns J. O., 1998, Science, 280, 400

Burns J. O., Eilek J. A., Owen F. N., 1982, in Heeschen D. S., Wade C. M., eds, IAU Symposium Vol. 97, Extragalactic Radio Sources. p. 45

Butenko A. V., Tyul'bashev S. A., 2016, Astronomy Reports, 60, 718

Capetti A., Massaro F., Baldi R. D., 2017a, AAP, 598, A49

Capetti A., Massaro F., Baldi R. D., 2017b, AAP, 601, A81

Capetti A., Massaro F., Baldi R. D., 2020a, Astronomy and Astrophysics, 633, A161

Capetti A., et al., 2020b, Astronomy and Astrophysics, 642, A107

Carilli C. L., Barthel P. D., 1996, AAR, 7, 1

Chakraborti S., Yadav N., Cardamone C., Ray A., 2012, Astrophysical Journal, Letters, 746, L6

Chambers K. C., et al., 2016, arXiv e-prints, p. arXiv:1612.05560

Chen G., Chen P., Shi Y., Hsieh C.-Y., Liao B., Zhang S., 2019, arXiv e-prints, p. arXiv:1905.05928

Chollet F., 2017, Deep Learning with Python, 1st edn. Manning Publications Co., USA

Clarke A. O., et al., 2017, Astronomy and Astrophysics, 601, A25

Colla G., et al., 1970, AAPS, 1, 281

Colla G., et al., 1972, AAPS, 7, 1

Colla G., et al., 1973, AAPS, 11, 291

Condon J. J., Cotton W. D., Greisen E. W., Yin Q. F., Perley R. A., Taylor G. B., Broderick J. J., 1998, AJ, 115, 1693

Cordey R. A., 1987, Monthly Notices of the RAS, 227, 695

Cotter G., Rawlings S., Saunders R., 1996, MNRAS, 281, 1081

Cotton W. D., et al., 2020, Monthly Notices of the RAS, 495, 1271

Cutri R. M., et al. 2013, VizieR Online Data Catalog, p. II/328

Dabhade P., Gaikwad M., Bagchi J., Pandey-Pommier M., Sankhyayan S., Raychaudhury S., 2017, Monthly Notices of the RAS, 469, 2886

Dabhade P., et al., 2020a, AAP, 635, A5

Dabhade P., et al., 2020b, Astronomy and Astrophysics, 635, A5

Dabhade P., et al., 2020c, Astronomy and Astrophysics, 642, A153

Delhaize J., et al., 2020, arXiv e-prints, p. arXiv:2012.05759

Delhaize J., et al., 2021, Monthly Notices of the RAS, 501, 3833

Di Matteo T., Allen S. W., Fabian A. C., Wilson A. S., Young A. J., 2003, Astrophysical Journal, 582, 133

Dieleman S., Willett K. W., Dambre J., 2015, Monthly Notices of the RAS, 450, 1441

Domínguez Sánchez H., et al., 2019, MNRAS, 484, 93

Douglas J. N., Bash F. N., Bozyan F. A., Torrence G. W., Wolfe C., 1996, AJ, 111, 1945

Duchi J. C., Hazan E., Singer Y., 2011, J. Mach. Learn. Res., 12, 2121

Duncan K. J., et al., 2019, Astronomy and Astrophysics, 622, A3

Edge D. O., Shakeshaft J. R., McAdam W. B., Baldwin J. E., Archer S., 1959, MemRAS, 68, 37

Ekers R. D., et al., 1989, Monthly Notices of the RAS, 236, 737

Elsmore B., Mackay C. D., 1969, MNRAS, 146, 361

Fanaroff B. L., Riley J. M., 1974, MNRAS, 167, 31P

Fanti C., Fanti R., Ficarra A., Padrielli L., 1974, AAPS, 18, 147

Fawcett T., 2006, Pattern Recognition Letters, 27, 861

Fomalont E. B., Petrov L., MacMillan D. S., Gordon D., Ma C., 2003, AJ, 126, 2562

Franzen T. M. O., et al., 2015, MNRAS, 453, 4020

Fukushima K., 1980, Biological Cybernetics, 36, 193

Garon A. F., et al., 2019, AJ, 157, 126

Gawroński M. P., Marecki A., Kunert-Bajraszewska M., Kus A. J., 2006, Astronomy and Astrophysics, 447, 63

Gawroński M. P., et al., 2010, MNRAS, 406, 1853

Gendre M. A., Wall J. V., 2008, MNRAS, 390, 819

Gendre M. A., Best P. N., Wall J. V., 2010, MNRAS, 404, 1719

Georgakilas G. K., Grioni A., Liakos K. G., Chalupova E., Plessas F. C., Alexiou P., 2020, Scientific Reports, 10, 9486

Gheller C., Vazza F., Bonafede A., 2018, MNRAS, 480, 3749

Ghisellini G., Tavecchio F., Foschini L., Ghirlanda G., 2011, Monthly Notices of the RAS, 414, 2674

Gillessen S., et al., 2012, Nature, 481, 51

Glorot X., Bengio Y., 2010, in Teh Y. W., Titterington M., eds, Proceedings of Machine Learning Research Vol. 9, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. PMLR, Chia Laguna Resort, Sardinia, Italy, pp 249–256, http://proceedings.mlr.press/v9/glorot10a.html

Glorot X., Bordes A., Bengio Y., 2011, in Gordon G., Dunson D., Dudík M., eds, Proceedings of Machine Learning Research Vol. 15, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. PMLR, Fort Lauderdale, FL, USA, pp 315–323, http://proceedings.mlr.press/v15/glorot11a.html

Goderya S. N., Lolling S. M., 2002, Astrophysics and Space Science, 279, 377

Goodfellow I., Bengio Y., Courville A., 2016a, Deep Learning. MIT Press

Goodfellow I. J., Bengio Y., Courville A., 2016b, Deep Learning. MIT Press, Cambridge, MA, USA

Gopal-Krishna Wiita P. J., 2000, Astronomy and Astrophysics, 363, 507

Górski K. M., Hivon E., Banday A. J., Wand elt B. D., Hansen F. K., Reinecke M., Bartelmann M., 2005, ApJ, 622, 759

Graham I., 1970, Monthly Notices of the RAS, 149, 319

Hahnloser R. H. R., Sarpeshkar R., Mahowald M. A., Douglas R. J., Seung H. S., 2000, Nature, 405, 947

Hanbury Brown R., Jennison R. C., Gupta M. K. D., 1952, Nature, 170, 1061

Hannun A., et al., 2014, arXiv e-prints, p. arXiv:1412.5567

Hao J., et al., 2010, ApJs, 191, 254

Hardcastle M. J., et al., 2019a, Monthly Notices of the RAS, 488, 3416

Hardcastle M. J., et al., 2019b, AAP, 622, A12

Harris A., 1972, Monthly Notices of the RAS, 158, 1

Harris A., 1973, Monthly Notices of the RAS, 163, 19P

Harwood J. J., 2017, Monthly Notices of the RAS, 466, 2888

Harwood J. J., Vernstrom T., Stroe A., 2020, Monthly Notices of the RAS, 491, 803

Hasanpour S. H., Rouhani M., Fayyaz M., Sabokrou M., 2016, arXiv e-prints, p. arXiv:1608.06037

Hastie T., Tibshirani R., Friedman J., 2001, The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA

He K., Zhang X., Ren S., Sun J., 2015, arXiv e-prints, p. arXiv:1512.03385

Heckman T. M., 1980, Astronomy and Astrophysics, 500, 187

Heckman T. M., Best P. N., 2014, Annual Review of Astron and Astrophys, 52, 589

Hine R. G., Longair M. S., 1979, Monthly Notices of the RAS, 188, 111

Ho L. C., 2009, Astrophysical Journal, 699, 626

Hodgkin A., Huxley A., 1952, Journal of Physiology, 117, 500

Hogbom J. A., Carlsson I., 1974, Astronomy and Astrophysics, 34, 341

Hota A., et al., 2011, Monthly Notices of the RAS, 417, L36

Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H., 2017, arXiv e-prints, p. arXiv:1704.04861

Hubel D. H., Wiesel T. N., 1968, The Journal of Physiology, 195, 215

Huertas-Company M., et al., 2015, Astrophysical Journal, Supplement, 221, 8

Hurley-Walker N., et al., 2015, Monthly Notices of the RAS, 447, 2468

Huynh M. T., Jackson C. A., Norris R. P., 2007, Astronomical Journal, 133, 1331

Intema H. T., Jagannathan P., Mooley K. P., Frail D. A., 2017, AAP, 598, A78

Ioffe S., Szegedy C., 2015, arXiv e-prints, p. arXiv:1502.03167

Ishwara-Chandra C. H., Saikia D. J., 1999, MNRAS, 309, 100

Jackson C., 2005, Publications of the Astron. Soc. of Australia, 22, 36

Jacobs C., Glazebrook K., Collett T., More A., McCarthy C., 2017, Monthly Notices of the RAS, 471, 167

Jansky K. G., 1933, Nature, 132, 66

Jarrett T. H., et al., 2017, ApJ, 836, 182

Jarvis M., et al., 2016, in MeerKAT Science: On the Pathway to the SKA. p. 6 (arXiv:1709.01901)

Jennison R. C., Das Gupta M. K., 1953, Nature, 172, 996

Jiang P., et al., 2019, Science China Physics, Mechanics, and Astronomy, 62, 959502

Johnston S., et al., 2007, Publications of the Astron. Soc. of Australia, 24, 174

Johnston S., et al., 2008, Experimental Astronomy, 22, 151

Jolliffe I. T., Cadima J., 2016, Philosophical Transactions of the Royal Society of London Series A, 374, 20150202

Jonas J., MeerKAT Team 2016, in MeerKAT Science: On the Pathway to the SKA. p. 1

Jones P. A., 1989, Proceedings of the Astronomical Society of Australia, 8, 81

Jones T. W., Owen F. N., 1979, Astrophysical Journal, 234, 818

Jurlin N., et al., 2020, Astronomy and Astrophysics, 638, A34

Kaiser C. R., Best P. N., 2007, Monthly Notices of the RAS, 381, 1548

Kaiser N., et al., 2002, in Tyson J. A., Wolff S., eds, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series Vol. 4836, PROCSPIE. pp 154–164, doi:10.1117/12.457365

Kaiser N., et al., 2010, in PROCSPIE. p. 77330E, doi:10.1117/12.859188

Kapińska A. D., et al., 2017, Astronomical Journal, 154, 253

Kembhavi A. K., Narlikar J. V., 1999, Quasars and active galactic nuclei : an introduction

Kempner J. C., Sarazin C. L., Markevitch M., 2003, Astrophysical Journal, 593, 291

Keysers D., Deselaers T., Gollan C., Ney H., 2007, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29, 1422

Kim Y., Jernite Y., Sontag D., Rush A. e. M., 2015, arXiv e-prints, p. arXiv:1508.06615

Kingma D. P., Ba J., 2014, arXiv e-prints, p. arXiv:1412.6980

Knapp A. W., 2006., Basic algebra:. Birkhauser,, Boston :

Komberg B. V., Pashchenko I. N., 2009, Astronomy Reports, 53, 1086

Komissarov S. S., 1988, Astrophysics, 29, 619

Kovalev Y. Y., Petrov L., Fomalont E. B., Gordon D., 2007, AJ, 133, 1236

Kowsari K., Heidarysafa M., Brown D. E., Jafari Meimandi K., Barnes L. E., 2018, arXiv e-prints, p. arXiv:1805.01890

Kozieł-Wierzbowska D., Stasińska G., 2011, MNRAS, 415, 1013

Koziel-Wierzbowska D., Goyal A., Zywucka N., 2020a, VizieR Online Data Catalog, p. J/ApJS/247/53

Kozieł-Wierzbowska D., Goyal A., Żywucka N., 2020b, Astrophysical Journal, Supplement, 247, 53

Krizhevsky A., Sutskever I., Hinton G. E., 2012a, in Pereira F., Burges C. J. C., Bottou L., Weinberger K. Q., eds, , Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp 1097–1105, http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Krizhevsky A., Sutskever I., Hinton G. E., 2012b, in Pereira F., Burges C. J. C., Bottou L., Weinberger K. Q., eds, , Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp 1097–1105, http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Kronberg P. P., Wielebinski R., Graham D. A., 1986, Astronomy and Astrophysics, 169, 63

Kuźmicz A., Jamrozy M., 2012, MNRAS, 426, 851

Kuźmicz A., Jamrozy M., Kozieł-Wierzbowska D., Weżgowiec M., 2017, Monthly Notices of the RAS, 471, 3806

Kuźmicz A., Jamrozy M., Bronarska K., Janda-Boczar K., Saikia D. J., 2018a, Astrophysical Journal, Supplement, 238, 9

Kuźmicz A., Jamrozy M., Bronarska K., Janda-Boczar K., Saikia D. J., 2018b, ApJs, 238, 9

Lacy M., Rawlings S., Saunders R., Warner P. J., 1993, Monthly Notices of the RAS, 264, 721

Lacy M., et al., 2020, PASP, 132, 035001

Lahav O., et al., 1995, Science, 267, 859

Laing R. A., 1994, in Bicknell G. V., Dopita M. A., Quinn P. J., eds, Astronomical Society of the Pacific Conference Series Vol. 54, The Physics of Active Galaxies. p. 227

Laing R. A., Riley J. M., Longair M. S., 1983, Monthly Notices of the RAS, 204, 151

Lal D. V., Rao A. P., 2005, Monthly Notices of the RAS, 356, 232

Lane W. M., Cotton W. D., van Velzen S., Clarke T. E., Kassim N. E., Helmboldt J. F., Lazio T. J. W., Cohen A. S., 2014, MNRAS, 440, 327

Langston G. I., Heflin M. B., Conner S. R., Lehar J., Carilli C. L., Burke B. F., 1990, Astrophysical Journal, Supplement, 72, 621

Lanusse F., Ravanbakhsh S., Mandelbaum R., Schneider J., Poczos B., 2017, in American Astronomical Society Meeting Abstracts #229. p. 342.05

Lara L., Cotton W. D., Feretti L., Giovannini G., Marcaide J. M., Márquez I., Venturi T., 2001a, AAP, 370, 409

Lara L., Márquez I., Cotton W. D., Feretti L., Giovannini G., Marcaide J. M., Venturi T., 2001b, AAP, 378, 826

Large M. I., Mills B. Y., Little A. G., Crawford D. F., Sutton J. M., 1981, MNRAS, 194, 693

Large M. I., Cram L. E., Burgess A. M., 1991, The Observatory, 111, 72

Law-Green J. D. B., Eales S. A., Leahy J. P., Rawlings S., Lacy M., 1995, Monthly Notices of the RAS, 277, 995

Le Cun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D., 1989, in Proceedings of the 2nd International Conference on Neural Information Processing Systems. NIPS'89. MIT Press, Cambridge, MA, USA, p. 396–404

LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D., 1989, Neural Computation, 1, 541

LeCun Y., Bottou L., Orr G., Müller K., 2012, in Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science. , doi:10.1007/978-3-642-35289-8_3

Leahy J. P., 1993, in Röser H.-J., Meisenheimer K., eds, Jets in Extragalactic Radio Sources. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 1–13

Leahy J. P., Williams A. G., 1984, Monthly Notices of the RAS, 210, 929

Leahy J. P., Bridle A. H., Strom R. G., 1996, in Ekers R. D., Fanti C., Padrielli L., eds, IAU Symposium Vol. 175, Extragalactic Radio Sources. p. 157

Lecun Y., Bottou L., Bengio Y., Haffner P., 1998a, in Proceedings of the IEEE. pp 2278–2324

Lecun Y., Bottou L., Bengio Y., Haffner P., 1998b, Proceedings of the IEEE, 86, 2278

Letawe G., Courbin F., Magain P., Hilker M., Jablonka P., Jahnke K., Wisotzki L., 2004, Astronomy and Astrophysics, 424, 455

Li B., Luo H., Zhang H., Tan S., Ji Z., 2017, arXiv e-prints, p. arXiv:1710.05477

Li X., Chen S., Hu X., Yang J., 2018, arXiv e-prints, p. arXiv:1801.05134

Liang H., Hunstead R. W., Birkinshaw M., Andreani P., 2000, Astrophysical Journal, 544, 686

Lin Y.-T., Shen Y., Strauss M. A., Richards G. T., Lunnan R., 2010, ApJ, 723, 1119

Lin M., Chen Q., Yan S., 2013, arXiv e-prints, p. arXiv:1312.4400

Longair M. S., Ryle M., Scheuer P. A. G., 1973, Monthly Notices of the RAS, 164, 243

Lonsdale C. J., et al., 2003, PASP, 115, 897

Lukic V., Brüggen M., 2019, Deep Learning in Radio Astronomy. Staats- und Universitätsbibliothek Hamburg, https://books.google.com.hk/books?id=USQ8zQEACAAJ

Lukic V., Brüggen M., Banfield J. K., Wong O. I., Rudnick L., Norris R. P., Simmons B., 2018, MNRAS, 476, 246

Lukic V., Brüggen M., Mingo B., Croston J. H., Kasieczka G., Best P. N., 2019, MNRAS, 487, 1729

Lundberg S., Lee S.-I., 2017, arXiv e-prints, p. arXiv:1705.07874

Ma Z., et al., 2019, ApJS, 240, 34

MacDonald G. H., Kenderdine S., Neville A. C., 1968, MNRAS, 138, 259

Machalski J., Jamrozy M., Zola S., 2001, Astronomy and Astrophysics, 371, 445

Machalski J., Koziel-Wierzbowska D., Jamrozy M., 2007, ACTAA, 57, 227

Machalski J., Kozieł-Wierzbowska D., Jamrozy M., Saikia D. J., 2008, Astrophysical Journal, 679, 149

Mackay C. D., 1969, MNRAS, 145, 31

Mackay C. D., 1971, MNRAS, 154, 209

Mahatma V. H., et al., 2019, Astronomy and Astrophysics, 622, A13

Maitra D. S., Bhattacharya U., Parui S. K., 2015, in Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR). ICDAR '15. IEEE Computer Society, USA, p. 1021–1025, doi:10.1109/ICDAR.2015.7333916, https://doi.org/10.1109/ICDAR.2015.7333916

Malarecki J. M., Jones D. H., Saripalli L., Staveley-Smith L., Subrahmanyan R., 2015, MNRAS, 449, 955

Mao M. Y., Johnston-Hollitt M., Stevens J. B., Wotherspoon S. J., 2009, MNRAS, 392, 1070

Mao M. Y., Sharp R., Saikia D. J., Norris R. P., Johnston-Hollitt M., Middelberg E., Lovell J. E. J., 2010, MNRAS, 406, 2578

Mauch T., Murphy T., Buttery H. J., Curran J., Hunstead R. W., Piestrzynski B., Robertson J. G., Sadler E. M., 2003, MNRAS, 342, 1117

McCarthy P. J., Kapahi V. K., van Breugel W., Persson S. E., Athreya R., Subrahmanya C. R., 1996, Astrophysical Journal, Supplement, 107, 19

Metcalf R. B., et al., 2019, AAP, 625, A119

Miley G. K., Perola G. C., van der Kruit P. C., van der Laan H., 1972, Nature, 237, 269

Miller A. S., 1993, Vistas in Astronomy, 36, 141

Mills B. Y., 1952, Nature, 170, 1063

Mills B. Y., 1981, Proceedings of the Astronomical Society of Australia, 4, 156

Mingo B., et al., 2019, MNRAS, 488, 2701

Mira J., Hernández F. S., eds, 1995, From Natural to Artificial Neural Computation, International Workshop on Artificial Neural Networks, IWANN '95, Malaga-Torremolinos, Spain, June 7-9, 1995, Proceedings Lecture Notes in Computer Science Vol. 930. Springer, doi:10.1007/3-540-59497-3, https://doi.org/10.1007/3-540-59497-3

Missaglia V., Massaro F., Capetti A., Paolillo M., Kraft R. P., Baldi R. D., Paggi A., 2019, A&A, 626, A8

Mitton S., 1970a, Astrophysics Letters, 5, 207

Mitton S., 1970b, Astrophysics Letters, 6, 161

Mitton S., 1970c, Monthly Notices of the RAS, 149, 101

Molina M., Bassani L., Malizia A., Bird A. J., Bazzano A., Ubertini P., Venturi T., 2014, Astronomy and Astrophysics, 565, A2

Moss A., 2018, arXiv e-prints, p. arXiv:1810.06441

Murgia M., et al., 2011, Astronomy and Astrophysics, 526, A148

Murphy T., Mauch T., Green A., Hunstead R. W., Piestrzynska B., Kels A. P., Sztajer P., 2007, Monthly Notices of the RAS, 382, 382

Murtagh F., 1988, Multivariate Analysis Methods: Background and Example. p. 308, doi:10.1007/3-540-50135-5_85

Murtagh F. D., Adorf H. M., 1992, in Data Analysis in Astronomy. pp 103–111

Nandi S., Saikia D. J., 2012, Bulletin of the Astronomical Society of India, 40, 121

Nilsson K., 1998, AAPS, 132, 31

Norris R. P., et al., 2011, PASA, 28, 215

Norris R. P., et al., 2013, Publications of the Astron. Soc. of Australia, 30, e020

Northover K. J. E., 1973, Monthly Notices of the RAS, 165, 369

Northover K. J. E., 1974, in Barbanis B., Hadjidemetriou J. D., eds, Galaxies and Relativistic Astrophysics. p. 61

O'Neill B. J., Jones T. W., Nolting C., Mendygral P. J., 2019, Astrophysical Journal, 887, 26

Ochsenbein F., Bauer P., Marcout J., 2000, AAPS, 143, 23

Odewahn S. C., Stockwell E. B., Pennington R. L., Humphreys R. M., Zumach W. A., 1992, Automated Star / Galaxy Discrimination with Neural Networks. p. 215, doi:10.1007/978-94-011-2472-0_28

Oquab M., Bottou L., Laptev I., Sivic J., 2014, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Ortega-Minakata R. A., Torres-Papaqui J. P., Andernach H., 2018, arXiv e-prints, p. arXiv:1808.09049

Owen F. N., 1993, Steps Toward a Radio H-R Diagram. p. 273, doi:10.1007/3-540-57164-7_104

Owen F. N., Ledlow M. J., 1994, in Bicknell G. V., Dopita M. A., Quinn P. J., eds, Astronomical Society of the Pacific Conference Series Vol. 54, The Physics of Active Galaxies. p. 319

Owen F. N., Rudnick L., 1976, ApJL, 205, L1

Pan S., Yang Q., 2010, IEEE Transactions on Knowledge and Data Engineering, 22, 1345

Parma P., de Ruiter H. R., Mack K. H., van Breugel W., Dey A., Fanti R., Klein U., 1996, AAPS, 311, 49

Paszke A., et al., 2019, in Wallach H., Larochelle H., Beygelzimer A., d Alché-Buc F., Fox E., Garnett R., eds, , Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp 8024–8035, http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pearson T. J., Readhead A. C. S., 1988, ApJ, 328, 114

Peng B., Chen R. R., Strom R., 2015, in Advancing Astrophysics with the Square Kilometre Array (AASKA14). p. 109 (arXiv:1501.00407)

Perley R. A., Butler B. J., 2017, ApJs, 230, 7

Petrillo C. E., et al., 2019, Monthly Notices of the RAS, 482, 807

Petrov L., Kovalev Y. Y., Fomalont E. B., Gordon D., 2006, AJ, 131, 1872

Piddington J. H., Minnett H. C., 1951, Australian Journal of Scientific Research A Physical Sciences, 4, 459

Planck Collaboration et al., 2016, AAP, 594, A13

Planck Collaboration et al., 2018, arXiv e-prints, p. arXiv:1807.06205

Pourrahmani M., Nayyeri H., Cooray A., 2018, Astrophysical Journal, 856, 68

Powers D., 2008, Mach. Learn. Technol., 2

Pratt L. Y., 1993, in Hanson S. J., Cowan J. D., Giles C. L., eds, , Advances in Neural Information Processing Systems 5. Morgan-Kaufmann, pp 204–211, http://papers.nips.cc/paper/641-discriminability-based-transfer-between-neural-networks.pdf

Prechelt L., 1996, in Orr G. B., Müller K.-R., eds, Lecture Notes in Computer Science, Vol. 1524, Neural Networks: Tricks of the Trade. Springer, pp 55–69

Prescott M., et al., 2018, Monthly Notices of the RAS, 480, 707

Proctor D. D., 2016, ApJS, 224, 18

Reber G., 1944, ApJ, 100, 279

Rengelink R. B., Tang Y., de Bruyn A. G., Miley G. K., Bremer M. N., Roettgering H. J. A., Bremer M. A. R., 1997, AAPS, 124, 259

Riley J. M., 1972, Monthly Notices of the RAS, 157, 349

Riley J. M., 1973, Monthly Notices of the RAS, 161, 167

Riley J. M., Branson N. J. B. A., 1973, Monthly Notices of the RAS, 164, 271

Robbins H., Monro S., 1951, Ann. Math. Statist., 22, 400

Robertson J. G., 1991, Australian Journal of Physics, 44, 729

Röttgering H., et al., 2011, Journal of Astrophysics and Astronomy, 32, 557

Ruder S., 2016, arXiv e-prints, p. arXiv:1609.04747

Rudnick L., Owen F. N., 1977, AJ, 82, 1

Rumelhart D. E., Hinton G. E., Williams R. J., 1986, Nature, 323, 533

Russakovsky O., et al., 2014, arXiv e-prints, p. arXiv:1409.0575

Russell S., Norvig P., 2009, Artificial Intelligence: A Modern Approach, 3rd edn. Prentice Hall Press, USA

Ryle M., 1952, Proceedings of the Royal Society of London Series A, 211, 351

Ryle M., Windram M. D., 1968, MNRAS, 138, 1

Ryle M., Smith F. G., Elsmore B., 1950, MNRAS, 110, 508

Sadler E. M., et al., 2002, Monthly Notices of the RAS, 329, 227

Sadler E. M., Ekers R. D., Mahony E. K., Mauch T., Murphy T., 2014, Monthly Notices of the RAS, 438, 796

Safouris V., Subrahmanyan R., Bicknell G. V., Saripalli L., 2009, MNRAS, 393, 2

Sahr B., Hunt G., Cornwell T., 2002, in Bohlender D. A., Durand D., Handley T. H., eds, Astronomical Society of the Pacific Conference Series Vol. 281, Astronomical Data Analysis Software and Systems XI. p. 160

Saikia D. J., Jamrozy M., 2009, Bulletin of the Astronomical Society of India, 37, 63

Saikia D. J., Konar C., Kulkarni V. K., 2006, Monthly Notices of the RAS, 366, 1391

Santurkar S., Tsipras D., Ilyas A., Madry A., 2018, arXiv e-prints, p. arXiv:1805.11604

Saripalli L., 2012, Astronomical Journal, 144, 85

Saripalli L., Hunstead R. W., Subrahmanyan R., Boyce E., 2005, AJ, 130, 896

Saxena A., et al., 2018, MNRAS, 475, 5041

Scaife A. M. M., Heald G. H., 2012, MNRAS, 423, L30

Schaefer C., Geiger M., Kuntzer T., Kneib J. P., 2018, Astronomy and Astrophysics, 611, A2

Schinnerer E., et al., 2007, Astrophysical Journal, Supplement, 172, 46

Schmidhuber J., 2014, arXiv e-prints, p. arXiv:1404.7828

Schödel R., et al., 2002, Nature, 419, 694

Schoenmakers A. P., Mack K. H., de Bruyn A. G., Röttgering H. J. A., Klein U., van der Laan H., 2000a, Astronomy and Astrophysics, Supplement, 146, 293

Schoenmakers A. P., de Bruyn A. G., Röttgering H. J. A., van der Laan H., Kaiser C. R., 2000b, Monthly Notices of the RAS, 315, 371

Schoenmakers A. P., de Bruyn A. G., Röttgering H. J. A., van der Laan H., 2001, AAP, 374, 861

Scoville N., et al., 2007, Astrophysical Journal, Supplement, 172, 1

Sebastian B., Ishwara-Chandra C. H., Joshi R., Wadadekar Y., 2018, Monthly Notices of the RAS, 473, 4926

Serra P., et al., 2015, MNRAS, 452, 2680

Seymour N., et al., 2008, Monthly Notices of the RAS, 386, 1695

Shabala S. S., Godfrey L. E. H., 2013, ApJ, 769, 129

Shabala S. S., Jurlin N., Morganti R., Brienza M., Hardcastle M. J., Godfrey L. E. H., Krause M. G. H., Turner R. J., 2020, MNRAS, 496, 1706

Shakeshaft J. R., Ryle M., Baldwin J. E., Elsmore B., Thomson J. H., 1955, MemRAS, 67, 106

Shannon C. E., Weaver W., 1949, The Mathematical Theory of Communication. University of Illinois Press, Urbana and Chicago

Shimwell T. W., et al., 2017, AAP, 598, A104

Shimwell T. W., et al., 2019, AAP, 622, A1

Skrutskie M. F., et al., 2006, AJ, 131, 1163

Smith F. G., 1952, Nature, 170, 1065

Solovyov D. I., Verkhodanov O. V., 2011, Astrophysical Bulletin, 66, 416

Solovyov D. I., Verkhodanov O. V., 2014, Astronomy Letters, 40, 606

Soltan A., 1982, Monthly Notices of the RAS, 200, 115

Sonntag D., Barz M., Zacharias J., Stauden S., Rahmani V., Fóthi Á., Lőrincz A., 2017, arXiv e-prints, p. arXiv:1709.01476

Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, J. Mach. Learn. Res., 15, 1929–1958

Stehman S. V., 1997, Remote Sensing of Environment, 62, 77

Stigler S., 1986, The History of Statistics: The Measurement of Uncertainty Before 1900. Belknap Series, Belknap Press of Harvard University Press, https://books.google.co.jp/books?id=M7yvkERHIIMC

Storrie-Lombardi M. C., Lahav O., Sodre L. J., Storrie-Lombardi L., 1992, in American Astronomical Society Meeting Abstracts. p. 65.08

Subrahmanyan R., Saripalli L., Hunstead R. W., 1996, Monthly Notices of the RAS, 279, 257

Szegedy C., et al., 2014, arXiv e-prints, p. arXiv:1409.4842

Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z., 2015, arXiv e-prints, p. arXiv:1512.00567

Szegedy C., Ioffe S., Vanhoucke V., Alemi A., 2016, arXiv e-prints, p. arXiv:1602.07261

Tamhane P., Wadadekar Y., Basu A., Singh V., Ishwara-Chandra C. H., Beelen A., Sirothia S., 2015, Monthly Notices of the RAS, 453, 2438

Tang H., Scaife A. M. M., Leahy J. P., 2019, Monthly Notices of the RAS, 488, 3358

Tang H., Scaife A. M. M., Wong O. I., Kapińska A. D., Rudnick L., Shabala S. S., Seymour N., Norris R. P., 2020, Monthly Notices of the RAS, 499, 68

Terni de Gregory B., Feretti L., Giovannini G., Govoni F., Murgia M., Perley R. A., Vacca V., 2017, A&A, 608, A58

Thompson A. R., Clark B. G., Wade C. M., Napier P. J., 1980, ApJs, 44, 151

Tremonti C. A., et al., 2004, ApJ, 613, 898

Venn K., et al., 2019, in Canadian Long Range Plan for Astronomy and Astrophysics White Papers. p. 5 (arXiv:1910.00774), doi:10.5281/zenodo.3755910

Venturi T., Bardelli S., Morganti R., Hunstead R. W., 1998, Monthly Notices of the RAS, 298, 1113

Waggett P. C., Warner P. J., Baldwin J. E., 1977, Monthly Notices of the RAS, 181, 465

Waldram E. M., Yates J. A., Riley J. M., Warner P. J., 1996, MNRAS, 282, 779

Waldram E. M., Pooley G. G., Davies M. L., Grainge K. J. B., Scott P. F., 2010, MNRAS, 404, 1005

Walmsley M., et al., 2020, Monthly Notices of the RAS, 491, 1554

Wayth R. B., et al., 2015, Publications of the Astron. Soc. of Australia, 32, e025

Wen Z. L., Han J. L., Liu F. S., 2012, ApJs, 199, 34

Werner P. N., Worrall D. M., Birkinshaw M., 1999, Monthly Notices of the RAS, 307, 722

Wiita P. J., Rosen A., Gopal-Krishna Saripalli L., 1989, Giant Radio Galaxies via Inverse Compton Weakened Jets. p. 173, doi:10.1007/BFb0036027

Williams W. L., et al., 2019, AAP, 622, A2

Williams M. J., Schoneveld L., Mao Y., Klump J., Gosses J., Dalton H., Bath A., Barnes S., 2020, Journal of Open Source Software, 5, 2314

Willis A. G., Strom R. G., Wilson A. S., 1974, Nature, 250, 625

Worrall D. M., Birkinshaw M., Cameron R. A., 1995, Astrophysical Journal, 449, 93

Wright E. L., et al., 2010, AJ, 140, 1868

Wu C., et al., 2019, MNRAS, 482, 1211

Xie S., Girshick R., Dollár P., Tu Z., He K., 2016, arXiv e-prints, p. arXiv:1611.05431

Yosinski J., Clune J., Bengio Y., Lipson H., 2014, arXiv e-prints, p. arXiv:1411.1792

Zeiler M. D., Fergus R., 2013, arXiv e-prints, p. arXiv:1311.2901

Zhang X.-G., Dultzin-Hacyan D., Wang T.-G., 2007, Monthly Notices of the RAS, 377, 1215

Zhang X., Wang Y., Zhang W., Sun Y., He S., Contardo G., Villaescusa-Navarro F., Ho S., 2019a, arXiv e-prints, p. arXiv:1902.05965

Zhang Q., Nicolson A., Wang M., Paliwal K. K., Wang C., 2019b, arXiv e-prints, p. arXiv:1912.12023

Zhu W. W., et al., 2014, Astrophysical Journal, 781, 117

Zou H., Gao J., Zhou X., Kong X., 2019, ApJS, 242, 8

de Bruyn A. G., 1989, Astronomy and Astrophysics, 226, L13

van Breugel W. J. M., Miley G. K., 1977, Nature, 265, 315

van Haarlem M. P., et al., 2013, AAP, 556, A2