Compact and Extended Radio Sources Classification using Deep Convolutional Neural Networks

Wathela Alhassan

August 2018

A project submitted in partial fulfilment of the requirements for the degree M.Sc. in the Department of Astronomy, as part of the National Astrophysics and Space Science Programme
UNIVERSITY OF CAPE TOWN

Supervisors: Prof. A.R. Taylor and Dr. Mattia Vaccari

Abstract

Upcoming surveys with new radio observatories such as the Square Kilometer Array will generate a wealth of imaging data containing large numbers of radio sources. Different classes of radio sources can be used as tracers of the cosmic environment, including the dark matter density field, to address key cosmological questions. Classifying these sources based on morphology is thus an important step toward achieving the science goals of next generation radio surveys. Extended Radio Sources have been traditionally classified as Fanaroff-Riley (FR) I and II, although some exhibit more complex 'bent' morphologies arising from environmental factors or intrinsic properties. In this work we present the FIRST Classifier, an on-line system for automated classification of Compact and Extended radio sources. We developed the FIRST Classifier based on a trained Deep Convolutional Neural Network Model to automate the morphological classification of compact and extended radio sources observed in the FIRST radio survey. Our model was trained independently for 20 times and achieved an average accuracy, precision, recall and F1 of 0.98. The current version of the FIRST classifier is able to identify the morphological class for a single source or for a list of sources as Compact or Extended (FRI, FRII and BENT).

Acknowledgements

My progress in my studies up to this present moment could not have been done without the guidance and support of my Supervisor Prof Russ Taylor and Dr Mattia Vaccari to whom I lend my appreciation.

I express my deepest appreciation to my parents, to my brothers to my sisters AND my friends who have been my constant source of inspiration during the preparation of this project work.

Plagiarism Declaration

I, Wathela Alhassan, know the meaning of plagiarism and declare that all of the work in the document, save for that which is properly acknowledged, is my own.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As noted by Sandage & Kristian (1975), the first step in studying a population of objects is a classification of those objects. The morphological classification of galaxies is an approach based on grouping them by their visual appearance. Studying radio galaxies on the basis of their morphology allows us to understand the formation and evolution of galaxies and their sub-components as a function of e.g. luminosity, environment, stellar mass and star formation rate over cosmic time (Helfand et al., 2015). Different classes of radio galaxies can be used as tracers of the cosmic environment, including the dark matter density field and galaxy clusters, to address key cosmological questions (Makhathini et al., 2015).

Over the next few years, deep wide-area surveys with new radio observatories such as the Karl G. Jansky Very Large Array (VLA), the Australian Square-Kilometre-Array Pathfinder, MeerKAT and eventually the Square Kilometre Array will be carried out, and a vast amount of radio images will become available. Manual inspection of these images will be impractical, which motivates developing tools that can automatically analyse them, including automated morphological classification techniques for radio sources.

The radio sky is populated by a variety of compact and extended sources. Compact sources are unresolved sources which have a single non-diffuse component. The overwhelming majority of radio sources at 1.4 GHz fluxes of 1 mJy or so are compact. Extended radio galaxies exhibit diffuse resolved emission, and have been traditionally classified using the Fanaroff-Riley (FR) scheme as FRI and FRII sources. Some extended sources also exhibit more complex bent morphologies arising from environmental factors or intrinsic properties and are thus often referred to as bent-tailed sources or simply bent sources. Bent sources can further be classified into two main classes based on how their jets appear into Wide-Angle Tail and Narrow-Angled Tail (NATs).

1.1 Radio Sky

1.1.1 Background

Radio astronomy was born in 1933 by Jansky (1933) when he discovered radio waves from extraterrestrial origins, Karl Jansky, working for Bell Laboratories, was tasked to determine the origin of a source of noise that was showing up in receivers operating in the 20 MHz region of the radio spectrum, he soon discovered that source of the noise was near the center of the milky way using a large directional receiving antenna.

Microwave receiving technology has grown rapidly during the second World War which led to the invention of the Microwave Radar, followed by important improvement on its systems included sensitive receivers, highly accurate steerable mounts, parabolic reflectors and low noise amplifiers. Soon after the War, these Radars have been modified and used largely for radio astronomy, where they discovered the meteor radio scattering, imaged the moon and performed the surveys of the sky at high and higher frequencies. During the 1950's Radio Astronomy became one of the mainstream branches of science (Hey, 1983).

The first radio map of the night sky was created and published in 1937 by Grote Reber using fully steerable parabolic antenna operating a VHF frequencies (Royal Astronomical Society of Canada. & Chant, 1988), followed by The First Cambridge Catalogue of Radio Sources (1C) (Ryle et al., 1950), then 2C (Shakeshaft et al., 1955), and 3C (Bennett & Simth, 1962) which opened the way for many other surveys being conducted soon after.

Radio telescopes generally function in a similar way as optical telescopes, using a parabolic reflector to collect energy (Radio waves) from a large area of the sky then reflect this energy to a detector (receiver). Radio telescopes have different focusses like Cassigrain, prime and offset. The resolution of the radio telescope means how close objects can be together and still be distinguished as separate objects, which can be measured by the Angular resolution. Angular resolution, θ, of a telescope depends on the diameter (D) of the dish and the wavelength (λ) at which the measurements are being made, mathematically can be represented as:

$$\theta \approx 1.22\frac{\lambda}{D} \tag{1.1}$$

This shows clearly that, the larger the telescope, the more detail can be observed in a given wavelength. For a given resolution, radio telescopes must be much larger than optical telescopes because the wavelengths of radio waves are so much larger than the wavelengths of visible light. To overcome the limitation of building a very large radio telescope and at the same time increase the collecting area and hence the resolution. Radio interferometry was the solution, where the radio telescopes are made in the form of arrays. In an array two or more telescopes are used and their signals are combined to produce a resolving power equivalent to a telescope with a diameter equal to their separation defers. The result was a significantly better resolution and

the detection of fainter sources.

The Very Large Array is one of the first large radio interferometers, constructed at altitude of 2124 meters above sea level on the Plains of San Agustin, west of Socorro, New Mexico, and fully operated in 1980. The VLA has 27 antennas arranged in a Y shaped configuration, each of which has a 25 meters diameter dish. Over the years, the VLA has been used to carry out a variety of wide-area continuum surveys, including the NRAO VLA Sky Survey (NVSS) (Condon et al., 1998) and the Faint Images of the Radio Sky at Twenty centimeters (FIRST, (Becker et al., 1995); (White et al., 1997)).

Over the next few years, deep wide-area surveys with new radio observatories such as the Karl G. Jansky Very Large Array (VLA), the Australian Square-Kilometre-Array Pathfinder (ASKAP, Johnston et al. (2007); Johnston et al. (2008)), MeerKAT (Jonas & MeerKAT Team, 2016) and eventually the Square Kilometre Array (SKA, Braun et al. (2015)) will be carried out, and a vast amount of radio images will become available. Should be noted that these telescopes will have similar resolution ($\approx 5^{"}$) to FIRST. Different projects are already planned to be executed using these instruments, for example, the Evolutionary Map of the Universe project (EMU, Hopkins et al. (2015)) which will use the new ASKAP telescope to detect about 70 million sources, and MeerKAT Large Area Synoptic Survey project (MeerKLASS, Santos et al. (2017)) which will use MeerKAT telescope to detect about 1 million sources. Manual inspection of these images will be impractical (Hocking et al., 2015), which motivates developing tools that can automatically analyse them, which includes automated morphological classification techniques for radio sources.

1.1.2 Galaxies

A galaxy is a gravitationally bound system of stars, stellar remnants, interstellar gas, dust, and dark matter. Most recent observations indicate that at least 2 trillion galaxies exist in the observable universe, Most of the galaxies are 10^2 to 10^4 parsecs in diameter and usually separated by distances on the order of millions of parsecs (Sparke & Gallagher, 2007). Galaxies can be grouped into a number of groups according to their morphology, size and other characteristics (Hubble & P., 1926; Buta, Ronald J., 2011)

- Morphology: Galaxies were classified into three main types: Ellipticals, Spirals and Irregulars.
- Size: Galaxies ranges from dwarfs with around a billion (10^9) stars to giants with around one hundred trillion (10^{14}) stars.
- Based on physical properties, interaction with other galaxies or the surrounded environment, and the probability of having a black hole at their active centres, galaxies were

categorized into: Star Forming Galaxies, and Active Galactic Nuclei, Starburst Galaxies, Catastrophic Rings and Interacting Galaxies.

Star-forming Galaxies (SFGs)

SFGs is relatively strong radio emitters hosted by spiral and irregular galaxies, characterized by steep GHz radio spectra ($\alpha_r \approx 0.7$) dominated by both, synchrotron emission and a flat free-free component predominantly at frequency $v \approx 30GHz$ (Searle & Sargent, 1972). SFGs synchrotron emission results from relativistic plasma accelerated in supernova remnants associated with massive ($M \gtrsim 8\mathrm{M}_{\odot}$) SF (Condon & J., 1992) or due to gas turning into luminous stars powered by nuclear reactions.

SFGs dominates the local ($z < 0.4$) radio luminosity function (LF) with power ranges between $P_{1.4GHz} \approx 10^{23} WHz^{-1}$ to $P_{1.4GHz} \approx 10^{24} WHz^{-1}$, and at high redshifts most galaxies are forming stars (redshift dependent)(Sadler et al., 2002). Figure 1.1 illustrates the sensitivity of today's premier multi-wavelength surveys to the Star Formation Rate (SFRs) of galaxies at high redshift.

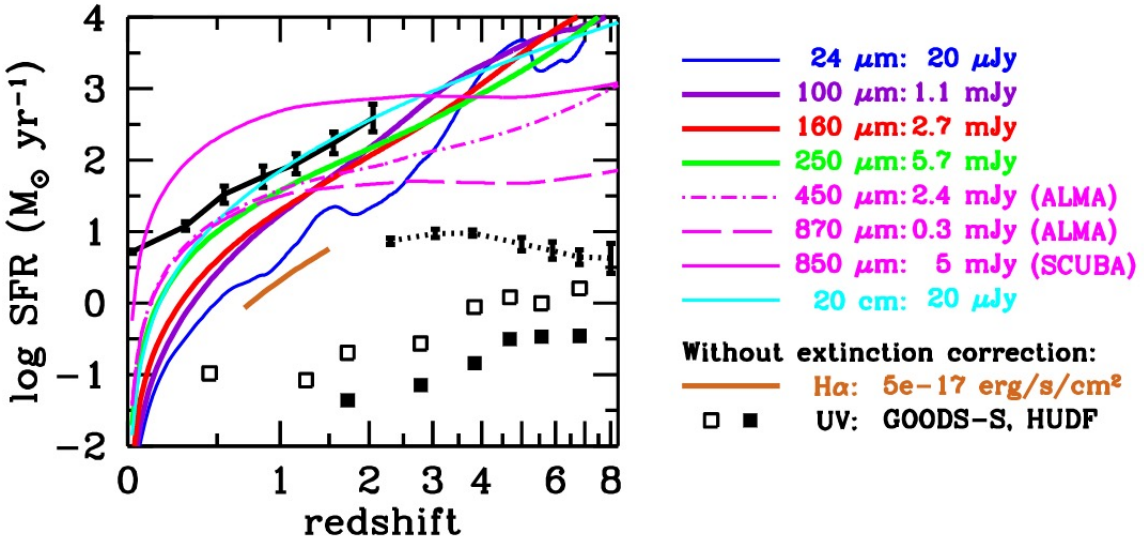


Figure 1.1: SFRs of galaxies at high redshift, where Rest-frame UV, IR, submillimeter and radio emission, as well as nebular lines such as Hα are all used to measure SFRs. Figure from (Madau & Dickinson, 2014)

In optical, SFGs are associated with the presence of blue stars which estimated through different techniques, the number of ionizing photons in $H\alpha$ observation are almost exclusively emitted by massive hot stars which have short life time, The evolution of their properties results from star-gas interactions, over cosmic time Galaxies evolve into populations with various colors, sizes, spectra, masses and morphologies (Guo et al., 2011).

Active Galactic Nuclei (AGN)

Seyfert (1943) first identified AGNs by their very broad emission lines observed in objects due to Doppler broadening. They are highly energetic phenomena, thought to be related to the presence of supermassive black holes at the centres of galaxies, whose emission cannot be attributed clearly and directly to stars (Aird et al., 2011). The main observational property of AGNs lies in their broad-band emission which can be detected throughout the radio, microwave, infrared, optical, ultraviolet, X-ray and gamma-ray part of the electromagnetic spectrum. AGNs are embedded within a host galaxy and can be detected at much higher redshift than normal galaxies, they often outshine their hosts by a large factor and thus display a nearly stellar shape, an example of AGN are NGC 1068 1 and NGC 4151. AGNs are traditionally divided into two main classes: Radio Loud (RL AGNs) and Radio Quiet (RQ AGNs).

RQ AGNs were first identified by Sandage & Allan (1965) after the discovery of the first quasar, 3C 273, a very strong radio source. He noticed that, there were many undetected similar sources in the sky. RQ AGN sources are associated with radio power of ≈ 3 magnitude smaller than their RL AGN counterparts, and they make up the majority ($> 90\%$) of the AGN class which makes them the norm, not the exception in radio sky (Padovani, 2011). Radio emission in RQ AGN is relatively weak, unlike in RL AGN, often spread across the host galaxy, and confined to the sub-kpc scale RL AGNs and RQ AGNs can be distinguished also through their emission, where RL AGN emitting a large fraction of their energy non-thermally and in association with powerful relativistic jets, while RQ AGN is dominated by thermal emission related to the accretion disk.

AGNs are further classified into different subclasses based on their spectroscopic and morphology properties : Radio Galaxies, Seyfert galaxies and Quasars. Our main focus in this work lies on Radio Galaxies and their radio Morphology classification which is discussed in the next section.

1.2 Radio Emission from Galaxies

Compact radio emission from galaxies (meaning unresolved at resolution of a few arcseconds) can arise from SFG or from compact emission from AGN that does not produce jets (i.e the AGN "core"). SFG have physical size about the same as the optical size of a galaxy, which is typically about 1" at reasonable redshift. This is why they are compact unresolved at resolution of a few arcseconds in radio.

Radio Galaxy (RGs) generally refers to the classical extended sources powered by jets from a central black hole. This terminology arises from the early days when these were the only radio emission known (they are the dominant type of the bright radio source population). Extended sources are thus the classical radio galaxies. Compact sources are AGN emission from AGN

cores or SF emission from SFGs.

1.2.1 Compact Sources

Compact sources are unresolved sources which have a single non-diffuse component associated with a flatter radio spectra, which are attributed to synchrotron self-absorption (Rybicki & Lightman, 2004). At different frequencies, different parts of compact sources become optically thick, resulting in a flattened integrated spectrum over a relatively large range in frequency (often called flat-spectrum sources). Generally compact sources associated with spectral index $\alpha_r = 0.5$ which is the clear way for distinguishing compact and extended sources (Massardi et al., 2010).

The overwhelming majority of radio sources ($\approx 80\%$) having fluxes $S_{1.4GHz} > 5mJy$ and redshift $z \lesssim 0.3$ are compact (Banfield et al., 2015; Lukic et al., 2018) or barely resolved at a radio resolution of 5 arcseconds (Baldi & Capetti, 2010) which implies no extended emission and no FR I/II division is possible. These type of sources have been recently classified as FR0 (which is a new entry) by Baldi et al. (2016) which were found to live in red massive early-type galaxies, with large black hole masses ($\gtrsim 10^8 M_\odot$), more core dominated (by a factor of ≈ 30) than FRIs while they share the same nuclear luminosity, and appears to be the dominant radio class of our local Universe.

1.2.2 Extended Sources

Extended radio sources are generally from AGN that produces jets and are usually elliptical. Their radio emission dominated by synchrotron radiation (relatively in high frequency) from interstellar relativistic electrons, which implies the existence of fast electrons moving in a magnetic field (Sadler et al., 1989). ERGs are optically thin and associated with a relativistic jets extending beyond the host galaxy (giant elliptical galaxy). Generally ERGs characterized by Giga Hertz (GHz) spectra of $\lesssim 4 \times 10^{20} WHz^{-1}$, emitting synchrotron radiation at relatively high frequencies , with a spectral index $\alpha_r \approx 0.7$ which considered as a signature of extended sources.

1.2.3 Morphological Classification of Extended Radio Sources

Extended radio sources have been traditionally classified using the Fanaroff-Riley (FR) scheme (Fanaroff & Riley, 1974) as FRI and FRII sources. FRIs and FRIIs are distinguished based on the position of low- and high-surface brightness regions in the extended components of the source. Some extended sources also exhibit more complex bent morphologies arising from environmental factors or intrinsic properties and are thus often referred to as bent-tailed sources or simply bent sources.

FRI Sources

FRI sources have smaller separation between the points of peak intensity in the two lobes, namely smaller than half the total extent of the source, and have the highest surface brightness along the jets and core (the edge-darkened FRIs) low-luminosity (Fanaroff & Riley, 1974). The environment of the sources appears to have an effect as well, since it has been found that, in rich clusters radio sources have higher probability of being FR Is, mainly because jets are more easily disrupted in dense environments (Gendre et al., 2013). Figure 1.2 shows an example of a FRI source.

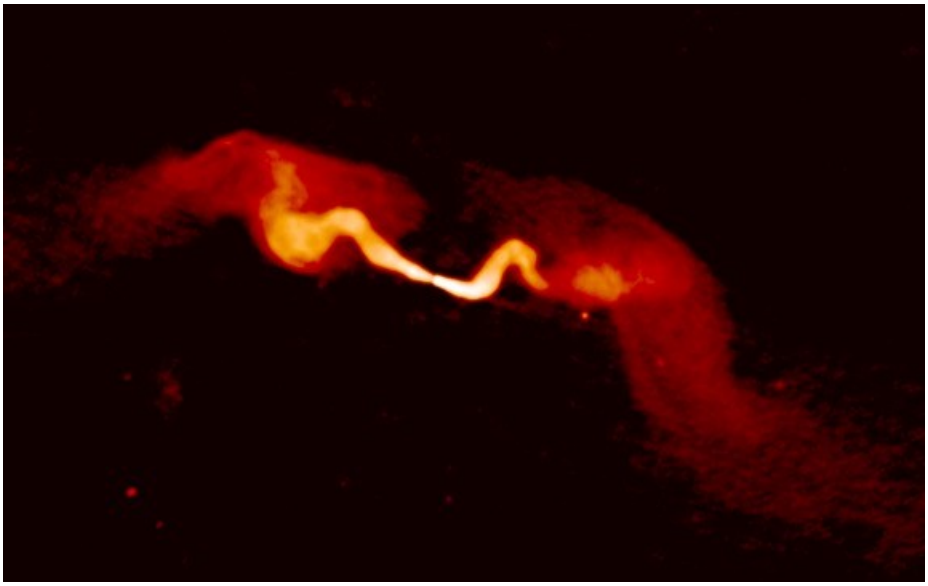


Figure 1.2: Radio Galaxy 3C31 (NGC 383) of FRI morphology type, at 1.4GHz using L band, RA: 01:07:24.96, Dec: 32:24:45.20 (J2000). Copyright NRAO 1996. Image from http://images.nrao.edu/AGN/Radio_Galaxies/328.

FRII Sources

In contrast to FRIs, FRIIs sources have a separation between the two points of peak intensity that is larger than half the total extent of the source and have the highest surface brightness at the edges (bright outer edges or edge-brightened FRIIs) with prominent hot spots (Fanaroff & Riley, 1974). FRIIs, unlike FRIs, can be found in more isolated environments and generally display stronger emission lines. Figure 1.3 shows an example of an FRII source.

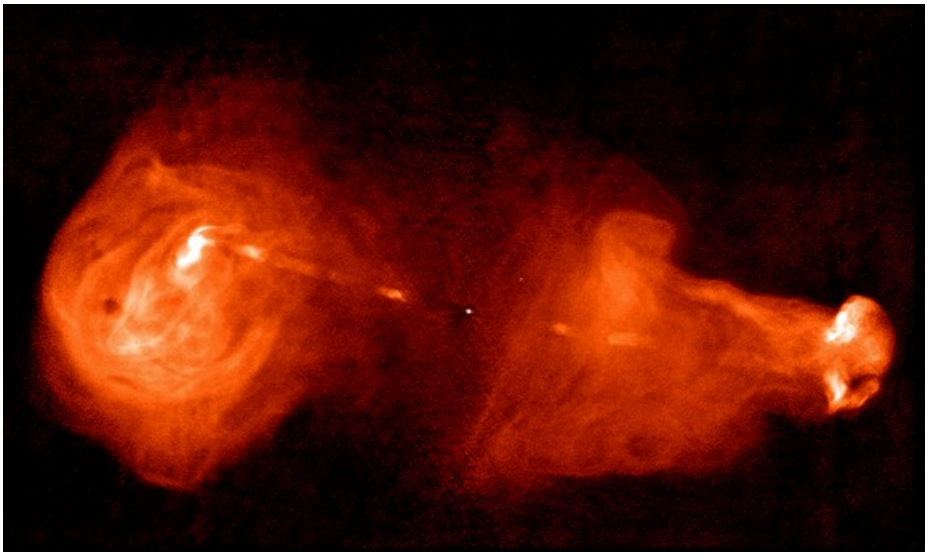
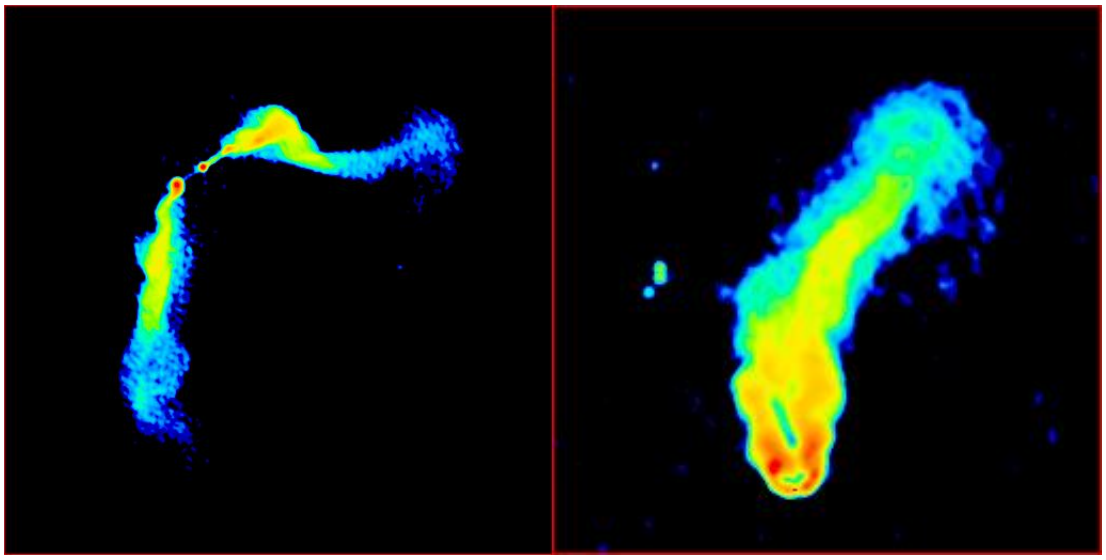


Figure 1.3: Radio Galaxy 3C353 of FRII morphology type, at 8.4 GHz using X band, RA: 17:20:28.16, Dec: 00:58:46.30 (J2000). Copyright NRAO 1996. Image from http://images.nrao.edu/AGN/Radio_Galaxies/284.

Bent-tailed Sources

Some extended sources exhibit more complex bent morphologies arising from environmental factors or intrinsic properties and are thus often referred to as bent-tailed sources or simply bent sources. Bent sources can further be classified into two main classes based on how their jets appear. Wide-Angle Tail (WATs) radio galaxies are sources where radio-emitting jets follow a wide C shape due to the dynamic pressure resulting from the host galaxy's rapid motion through the surrounding intracluster medium (ICM) (Sakelliou & Merrifield, 2000), located at or close to host cluster's center with higher peculiar velocities (Douglass et al., 2007, 2011).

Narrow-Angled Tail (NATs) radio galaxies are sources where the source resides in the cluster's outer regions with larger peculiar velocities and distinguished by its diffuse tail that follows a narrow C shape due to the host galaxy's rapid motion through the ICM, at higher resolution its tail can often be seen splitting up into two tails (Owen & Laing, 1989). NATs and WATs are also called - due to their bright head - Head Tail (HT) sources in the literature (Proctor, 2011). An example of WAT and NAT sources are shown in Figure 1.4. Because Bent sources are mostly located in clusters can be used to trace clusters at higher redshifts, especially when information from other wavelengths (e.g. optical or X-ray) is not available (Blanton et al., 2000, 2001; Blanton & Gregg, 2003).



(a) 3C 465 (b) 3C 83.1B

Figure 1.4: a) Radio Galaxy 3C 465 (NGC 7720) of WAT morphology type, at $1414MHz$, b) Radio Galaxy 3C31 (NGC 383) of NAT morphology type, at $1380MHz$. Copy right NRAO 1996. Image from http://www.cv.nrao.edu/~abridle/images.htm.bRadio.

1.3 Machine Learning and Deep Learning in Astronomy

Over the last few years, different machine learning and deep learning techniques have been successfully applied to astronomical applications, on both images and spectral data (Lukic & Brüggen, 2016). For example distinguishing between normal and peculiar optical galaxies by training Artificial Neural Networks (ANNs) (Naim et al., 1997). Estimation of the photometric redshift of galaxies using ANNs (Collister & Lahav, 2004), improved methods of the estimation of the photometric redshift of galaxies with more accurate results (Zhang et al., 2009; Bonfield et al., 2009) and most recent improvement by D'Isanto & Polsterer (2018).

The application of artificial neural networks to the problem of optical galaxy morphology classification has been the subject of active work since the early nineties (Storrie-Lombardi et al., 1992; Lahav et al., 1995; de la Calleja & Fuentes, 2004). In astronomy, projects such as Galaxy Zoo (Lintott et al., 2008) have generated strong interest in applying convolutional neural networks to visually classified galaxy samples (Dieleman et al., 2015). Recent examples are de la Calleja & Fuentes (2004) where neural network and locally weighted regression were used for morphological classification, Selim et al. (2016) and Selim & Abd El Aziz (2017) where a method based on non-negative matrix factorization algorithm were used to classifiy the galaxy morphology. Similar method has also been applied to the galaxy-star classification (Kim & Brunner, 2017).

Most of machine learning exploitation in astronomy has been done with optical data, little

work has been done on the morphological classification of radio galaxies. Unsupervised radio source classification has been performed using the Self Organizing Kohonen Map dimensionality reduction technique (Polsterer et al., 2015; Polsterer et al., 2015), which combines and sorts similar sources into classes and produces a single template representation of every class.

An application of Convolutional Neural Networks to extended radio galaxy morphology was presented by Aniyan & Thorat (2017), where they classified extended radio sources into three types, FRI, FRII and Bent, with an average precision of 91%, 75% and 95% respectively. In this work we have extended the problem to include compact sources using a similar dataset with different convolutional neural network model structure and data augmentation techniques.

Following the developments in optical galaxy morphology classification, the Radio Galaxy Zoo project (Banfield et al., 2015) has recently engaged many citizen scientists to identify the morphological type of radio sources and determine their host galaxy by combining infrared and radio observations. However, their classification scheme is based on the number of components of extended radio sources and does not lend itself to be directly interpreted in terms of FRIs, FRIIs and bent sources. However, Lukic et al. (2018) have applied convolutional neural networks to the classification of sources according to this scheme and achieved a final test accuracy of 94.8 per cent on Radio Galaxy Zoo Data Release 1.

1.4 Objective

The main purpose of this work is to automate the morphological classification of compact and extended radio sources (three classes FRI, FRII and BENT (WAT and NAT)) by developing a classifier that uses a trained deep Convolutional Neural Network (CNN) model to generate accurate and robust predictions.

1.5 Contribution of the Dissertation

In this dissertation, we have developed the FIRST Classifier (an open source tool) based on a trained Deep Convolutional Neural Network Model to automate the morphological classification of compact and extended radio sources observed in the FIRST radio survey. The software has a fully automated pipeline that, retrieves and cleans radio images, extracts and classify the target source and visualize the results. The FIRST Classifier is publicly available at: https://github.com/wathela/FIRST-CLASSIFIER

This work has been published in MNRAS and available at: https://academic.oup.com/mnras/article/480/2/2085/5060783

1.6 Structure of the Thesis

This Thesis is structured as follows:

- **Chapter (1) - Introduction** An introduction of radio astronomy and the radio sky components, Galaxies and their emission in radio wavelength and the morphological classification of radio sources are provided. A brief overview on machine learning and deep learning applications in astronomy is also presented.
- **Chapter (2) - Data** The surveys and datasets used in this work as well as preprocessing and data augmentation techniques are described.
- **Chapter (3) - Deep Learning** A brief background of deep learning is presented including developments in the field over time, advantages, disadvantages and most popular tools libraries.
- **Chapter (4) - Convolutional Neural Networks** An introduction of the Convolutional Neural Networks (CNN) and its layers were described. Training the CNN, advantages, disadvantages and achievements were also presented.
- **Chapter (5) - Experiments** We built 3 different CNN models for the classification of extended and compact radio sources, description of each model is presented.
- **Chapter (6) - Results** Comparison of the 3 models results is presented and discussed, metrics used for evaluation are also explained.
- **Chapter (7) - FIRST Classifier** We developed an automatic classifier based on a CNN model for in FIRST survey sources, the Features of the classifier and input-output examples are presented.
- **Chapter (8) - Conclusion and Future Work** A summary and main remarks of the work are presented. Suggestions for future work including developing of the current work are also discussed.

Chapter 2

Data

In this section I describe the datasets that were used in the study. In order to train any Convolutional Neural Network Model for a supervised classification problem, we need labelled dataset which means, in our case, pre-classified radio sources. Unfortunately, in the literature there are very few and limited classified catalogues for radio sources. In this study, we made use of all the available catalogues to the best of our knowledge. It must be noted that for Compact sources, we found FR0 catalogue (Baldi et al., 2018) after the work has been begun so we did not include it in our catalogues, rather we used it to test the performance of our classifier in chapter 7. For our 4 classes, FRI, FRIIs, Bent and Compact, we have formed separate samples from different catalogues and retrieved their respective images from FIRST images archive data. Due to our limited images, an augmented set were created by rotating and flipping the original images in order to have enough images of our four classes to train the models.

2.1 Radio Surveys

2.1.1 The FIRST Survey

The Faint Images of the Radio Sky at Twenty centimeters survey (FIRST, Becker et al. (1995)), begun in the spring of 1993 using the VLA at 1.4 GHz with 5" angular resolution and a 1 mJy/beam sensitivity. The first released catalogue of the first observation sessions has 138,665 radio sources derived from the initial 1550 deg^2 of the north Galactic cap, with source position's accuracy better than $1''$ and the flux density scale is accurate up to 5% (White et al., 1997). The final released catalogue of FIRST (Helfand et al., 2015) covered 10,575 square degrees of sky with 8,444 square degrees in the north and 2,131 square degrees in the south, over a total of eleven observing sessions in the B-configuration and 4000 hours of observing time, Figure 2.1 shows the time allocated and the sky coverage achieved over the 18 years of survey observations. The survey produced a 1.4 GHz source catalogue with flux densities, subarcsecond positions,

and morphological information for 946,432 sources . Both the northern and southern areas were chosen to coincide approximately with the area covered by the Sloan Digital Sky Survey (SDSS, York et al. (2000)).

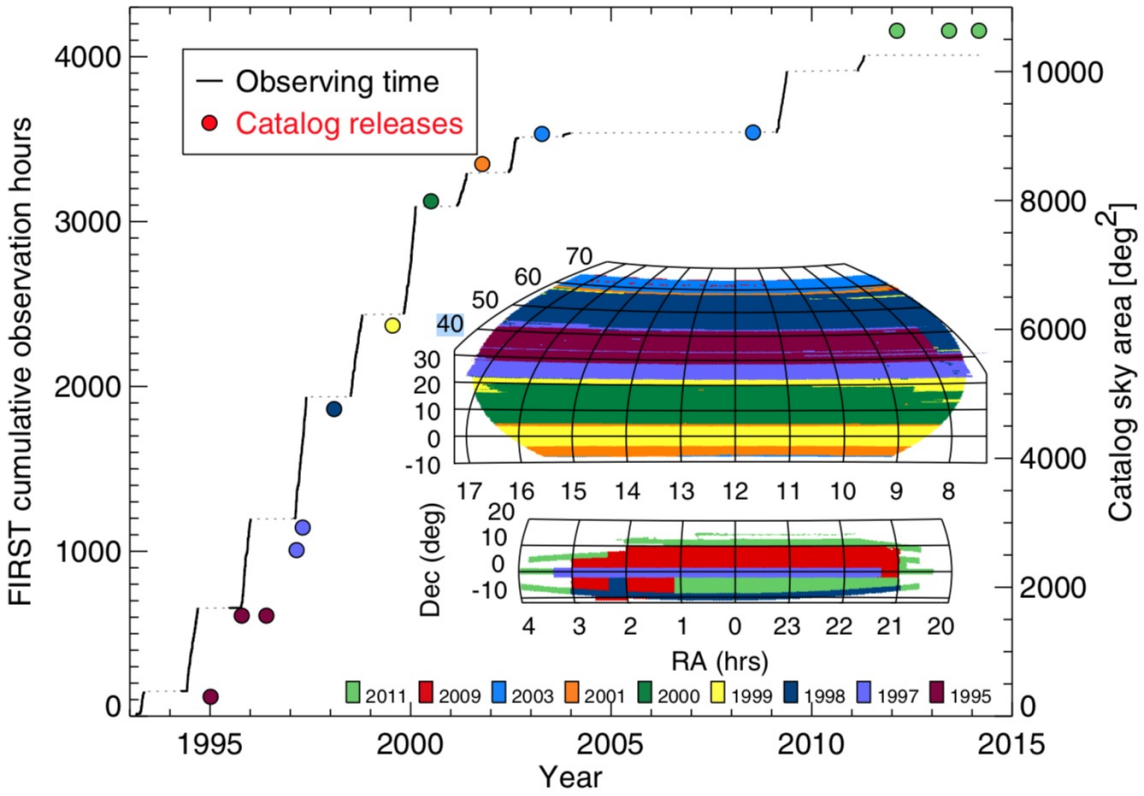


Figure 2.1: The figure shows the sky coverage of the *FIRST* survey by the catalouge release dates as a function of time (for 20 years). Sky area in square degree is shown at the right axis while the cumulative observing time is shown at the left axis. The total coverage area of the sky is $10,575$ deg^2 with total of $4009hr$ of VLA observing time. Figure from (Helfand et al., 2015)

2.1.2 The NVSS Survey

The NRAO VLA Sky Survey (NVSS) (Condon et al., 1998) contains 2×10^6 discrete sources with a flux density higher than 2.5 mJy continuum cubes, the survey was carried in three years out between 1993 and 1996 and covers the sky at $\delta > -40^o$ of J2000.0 using the compact D and C configurations of the VLA. The catalogue consists of 271,466 independent continuum intensity and linear polarization images at 1.4 GHz. All the images in the catalouge have $\theta = 45^{"}$ FWHM resolution and nearly uniform sensitivity. Gaps in the survey for a certain areas of the sky were observed and added to the catalogue in 1998.

2.2 Classified Radio Catalogues

We constructed our sample of compact extended radio sources from four catalogues, each of which contains the source coordinates and their classification label, in this section we briefly describe the catalogues and their relevant number of sources that we used from each.

2.2.1 FRICAT Catalogue

For FRIs we used the FRICAT catalogue by Capetti et al. (2016), which merges data from NRAO VLA Sky Survey (NVSS, Condon et al. (1998)), Faint Images of the Radio Sky at Twenty Centimeters (FIRST, Becker et al. (1995)) and the Sloan Digital Sky Survey (SDSS, York et al. (2000)). FRICAT consists of 219 FRI galaxies with redshifts ≤ 0.15. All the sources included here have an edge-darkened radio morphology with radius extending larger than 30 kpc from the host.

2.2.2 FRIICAT Catalogue

For FRIIs we used the FRIICAT catalogue (Capetti & et al, 2017), which like the previous catalog contains samples that were obtained by merging observations from FIRST and NVSS. FRIICAT consists of 122 FRII galaxies and contains sources that have an edge-brightened radio morphology with redshifts ≤ 0.15.

FRICAT and FRIICAT are essentially a subset of the catalogue of 18,286 radio sources built by Best & Heckman (2012) as a response to the shortage of FRI sources in the literature and to study the main properties of FRI and FRII galaxies based on their spectroscopic classification. They used FIRST images for their morphological classification at FIRST angular resolution (5"), and all sources were chosen to have a redshift $z \leq 0.15$. For a source to be classified as an FRII, it must have emission peaks at least 30 kpc from the optical host center. Each of the three authors of FRIICAT performed this inspection for each source independently and they only included sources where at least two of them agreed that it was a FRII.

2.2.3 Proctor Catalogue

In order to create a reliable catalogue of bent sources, we made use of the Proctor (2011) catalogue. The catalogue has examined and separated objects in the FIRST catalogue using different pattern recognition techniques and visual inspection, into sources with more than four components (lower-count membership i.e. singles, doubles, triples) groups and sources with more than four components (expected to have complex morphology) groups. They classified the groups with four and more members (7016 groups) into different bent types including WAT, NAT and Compact. To define our bent-tailed (hereafter BENT) sample we only used the confirmed WATs and NATs, which amount to 192 sources.

2.2.4 CoNFIG Catalogue

For compact (hereafter COMP) sources, we made use of the Combined NVSS-FIRST Galaxy catalogue (CoNFIG, Gendre & Wall (2009); Gendre et al. (2010)), which include new VLA observations, optical identifications and redshift estimates of Compact, FRI and FRII sources. The catalogue consists of 859 sources over 4 samples (CoNFIG-1, 2, 3 and 4 with flux density limits of S_1.4GHz = 1.3, 0.8, 0.2 and 0.05 Jy respectively). It is 95.7% complete in radio morphology classification and 74.3% of the sources have redshift data. All the sources smaller than 3 arcsec (unresolved) were classified as compact sources. We also made use of FRIIs sources in this catalogue and added them to the FRIICAT sources. The number of sources of each type in our sample is summarised in Table 2.1.

2.2.5 Image Pre-Processing and Data Augmentation

We retrieved the FIRST images for the catalogue described in the previous section via the online FIRST image archive as FITS files. Images were first cleaned, rescaled and cropped to reduce the dimensionality of the input and then saved as PNG files. It was useful to crop the images because the object of interest is in the middle of the image with a large amount of sky background, and to reduce the number of features to be extracted. We then rescaled the images to speed up training, with little to no effect on predictive performance. Images were cropped from 300×300 pixels to 150×150. For a small number of images, where an extended source was extremely large, the cropping operation removed part of the sources. We thus removed 84 sources from our dataset.

For the cleaning process we used the same method adopted by Aniyan & Thorat (2017), where all the pixels values below 3σ were removed (set to zero) in order to remove the background noise. Due to the small number of labelled images, artificial images were created by flips and rotations, to generate sufficient data for training our model. Every labelled image was rotated by a random angle, then flipped along the x-axis to produce the artificial images. Flips and rotations do not increase the topological information contained within the data, but alter the orientation of the object, Figure 2.2 illustrates the pre-processing steps.

These operations have been carried out on every image from our original dataset. Must be noted that, the original dataset (837 images) was divided first into three groups (Training, testing and validating) then augmented (flipped and rotated) to make sure the model will not be trained and tested on the same images. We also made sure that we have equal number of augmented images for each class (FR I, FR II, Bent and Compact) so our model won't be biased to a certain class. Table 2.1 details the number of our original images, the original and the Flipped/Rotated version of training, testing and validation of our final dataset. Sample images of FRI, FRII, BENT and COMP sources are shown in Figure 2.3, 2.4, 2.5 and 2.6 respectively, after being pre-processed. The augmented dataset, along with the corresponding class labels,

were then used to train our CNN model for the purpose of Compact and Extended radio galaxies morphology classification. As shown in Table 2.1, the augmented dataset was divided into three parts. The first part was used to train the model (hereafter train) which represents 60% of the whole data. The second and third parts were test and validation datasets which each represents 20% of the whole datasets. This approach follows a common practice in the field to train the model on larger portion of the data and use the remaining for testing and validating the model which is usually $60 - 20 - 20$ ratio respectively for training, test and validation sets.

Type	Original Sample	Train		Test		Val	
		ORG	F/R	ORG	F/R	ORG	F/R
COMP	121	73	3,445	24	1000	24	1000
FRI	201	121	3,445	40	1000	40	1000
FRII	338	204	3,445	67	1000	67	1000
BENT	177	107	3,445	35	1000	35	1000
Total	837	505	13,780	166	4000	166	4000

Table 2.1: Number of: original sample images, original (ORG) and flipped/rotated (F/R) version images for training, testing and validation images.

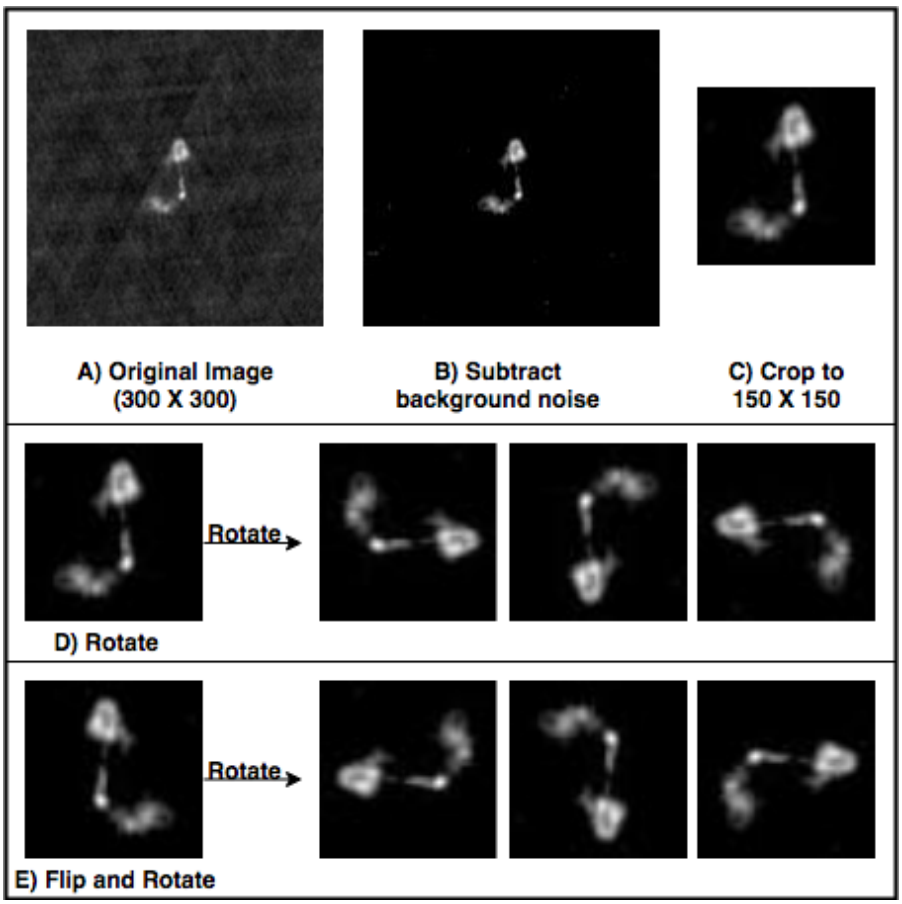


Figure 2.2: Pre-processing steps of the images

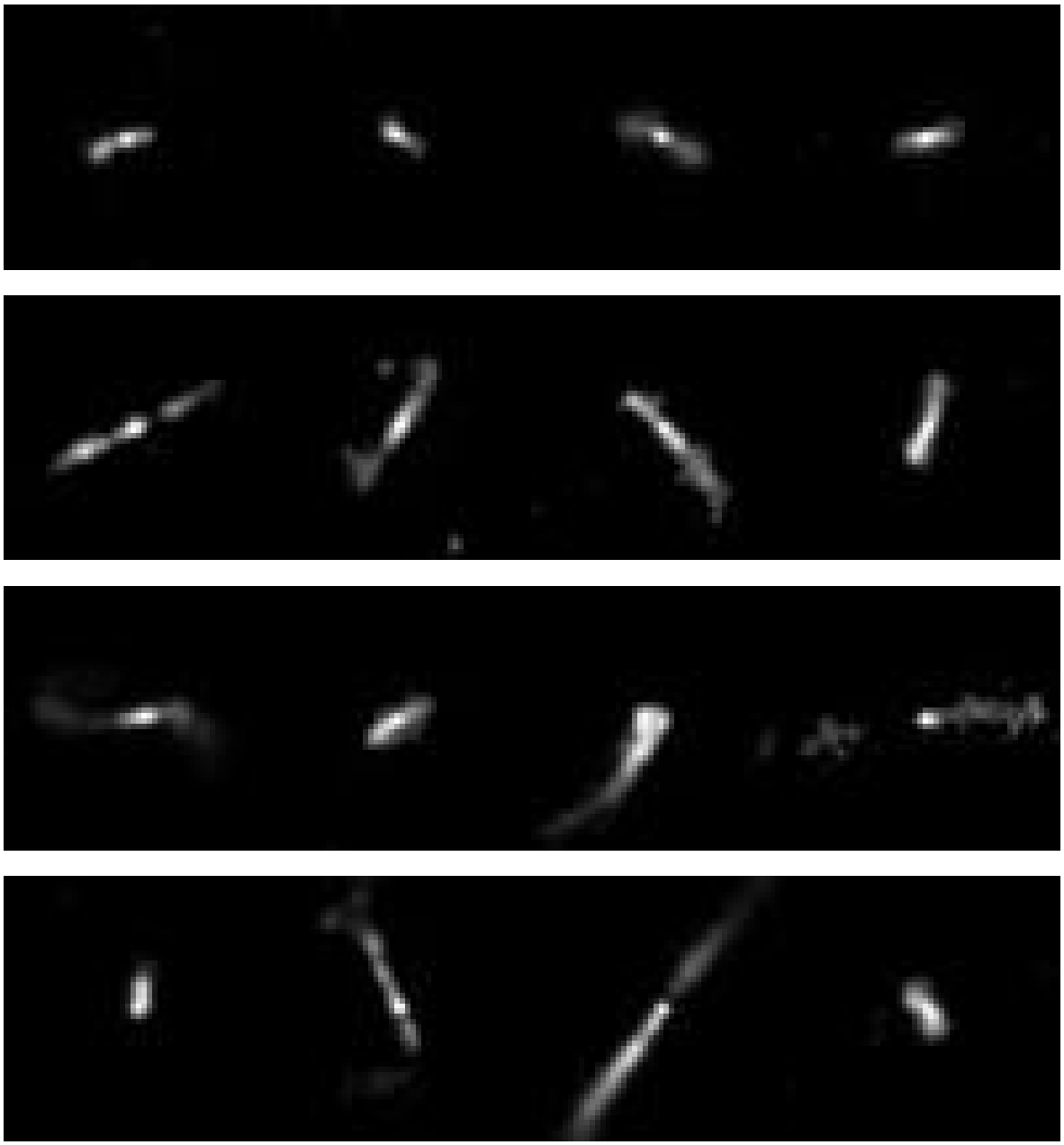


Figure 2.3: FRI galaxies sample images

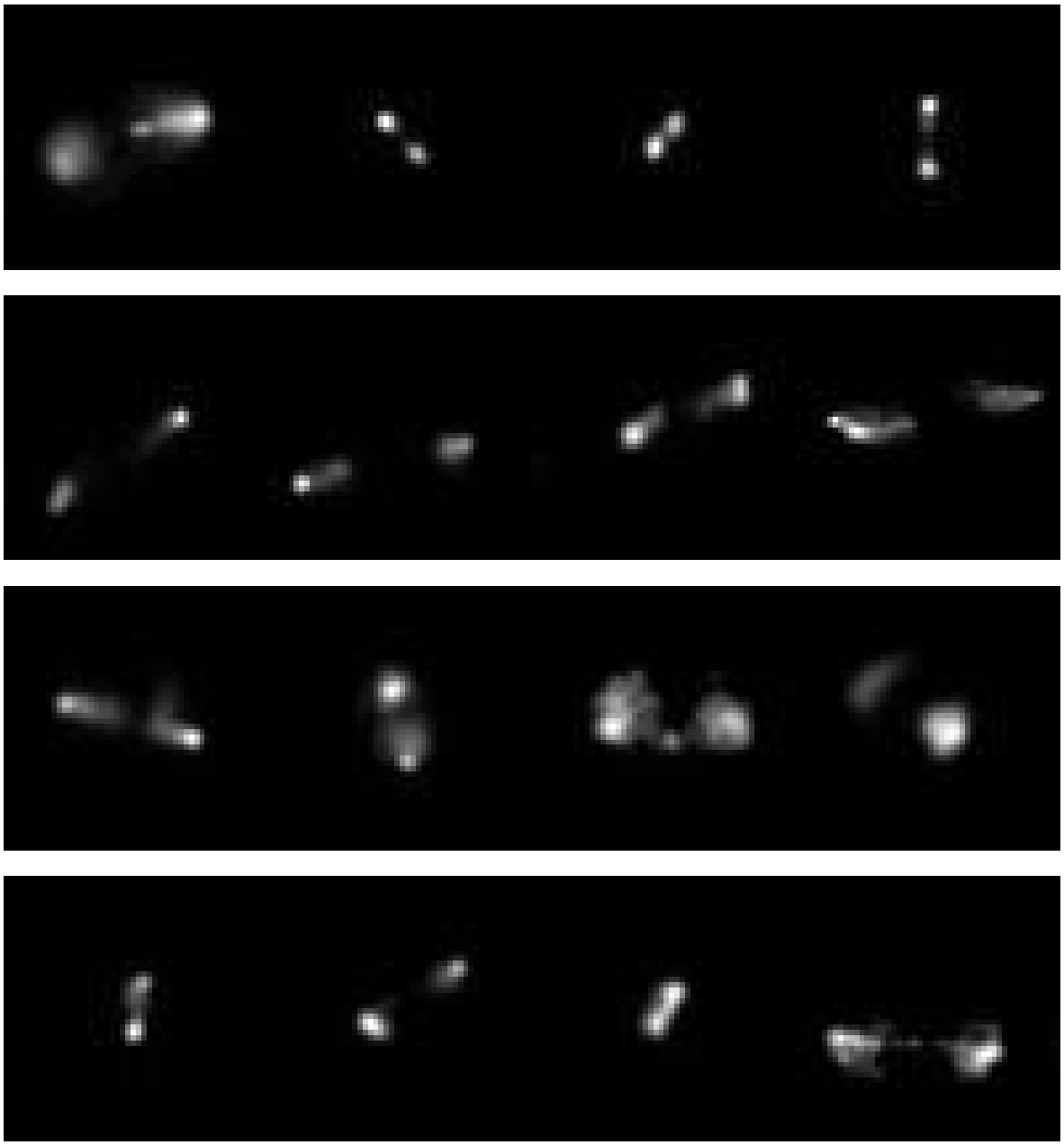


Figure 2.4: FRII galaxies sample images

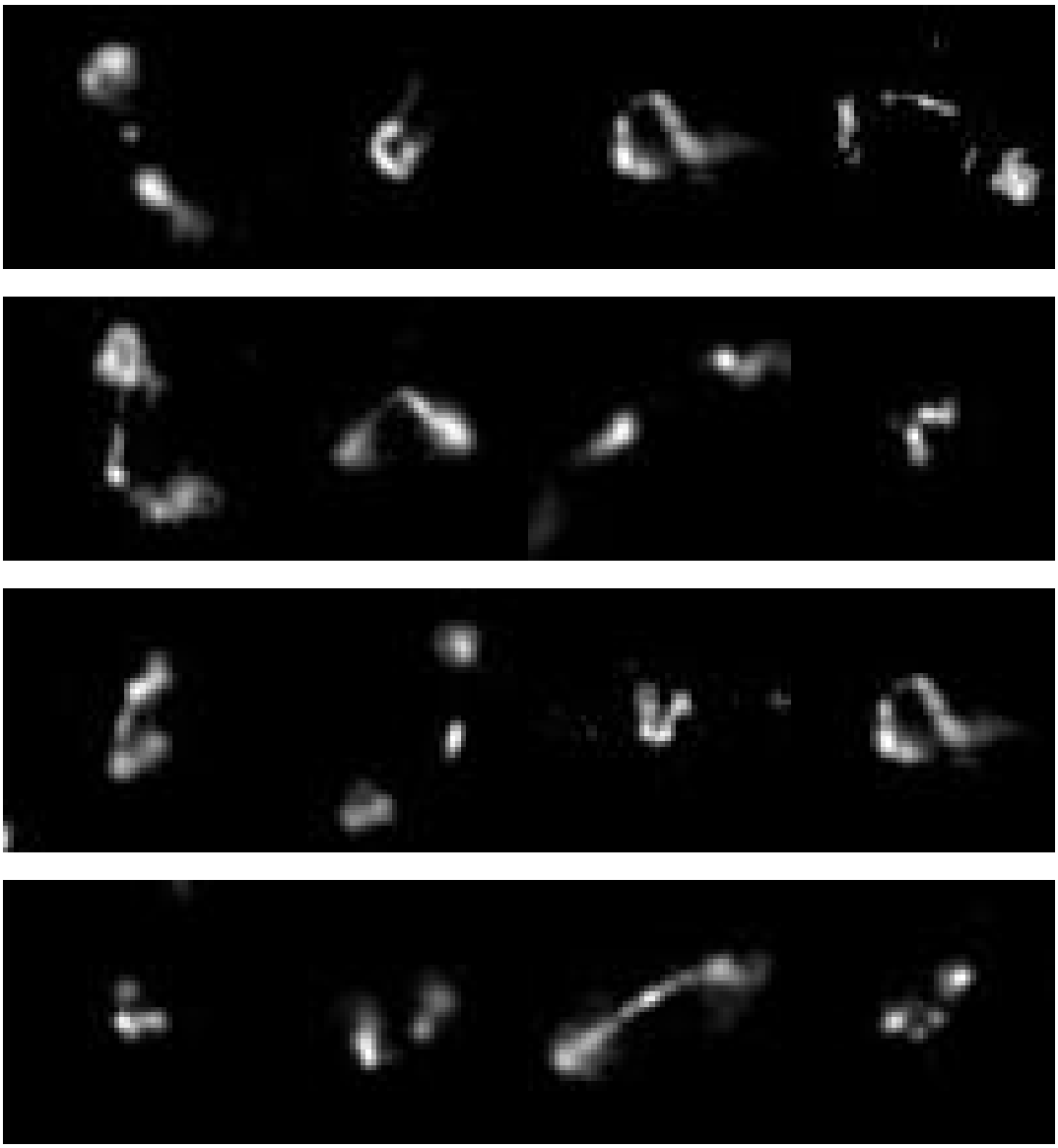


Figure 2.5: BENT galaxies sample images (WATs and NATs)

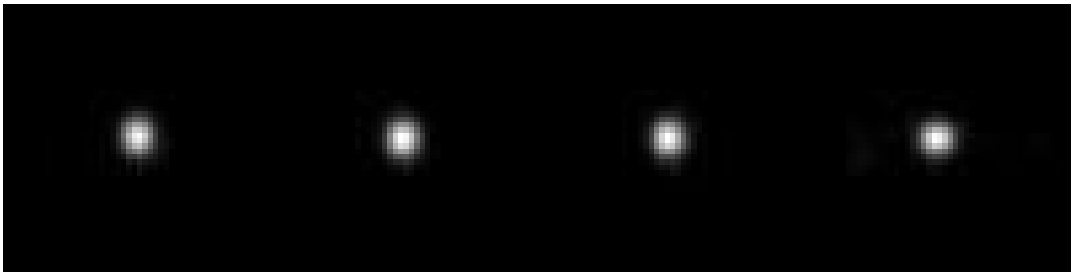


Figure 2.6: Compact galaxies sample images.

Chapter 3

Deep Learning

Deep Learning enables learning representation of data with multiple levels of abstraction using neural network models consisting of multiple processing (hidden) layers. Historically, the computations being performed inside the neural networks were computationally expensive and required a lot of data, which made it impractical due to the limitation of computational power and data. In the last decade Deep Learning has become popular due to the advancements in computer hardware (CPUs became much faster and GPUs were significantly improved) and the exponential growth of data in different fields. These advancements make it possible to train very deep (larger number of hidden layers) and highly parametrized neural networks, which have become known as Deep Neural Networks (DNNs).

DNNs allow for very complex representations of data to be modelled, which opens the door to analyzing high-dimensional data (i.e. images, videos), unlike traditional machine learning approaches, where features of the data need to be manually defined prior to modelling. This ability of DNNs has dramatically improved the state-of-the-art visual object recognition, object detection, speech recognition and many other domains.

3.1 Background

Deep Learning is a sub-field of Machine Learning focusing on statistical models making use of Deep Neural Networks (DNN). Unlike traditional machine learning algorithms where data-specific features must be extracted manually , DNN has the ability to generate and learn complex and hierarchical features (representations) from raw data without human intervention, giving an effective tool for automatic feature extraction from large datasets which require domain expertise and consume a lot of time especially with current world data (Goodfellow et al., 2016). For the problem of classifying radio sources, due to the large amount of radio images being delivered by different radio surveys, manual extraction of features is almost impossible which makes deep learning a good option for tackling the task.

The history of deep learning goes back 1943, when a neurophysiologist Warren McCulloch and mathematician Walter Pitts developed a computer model based on the resemblance of the neural networks of human brain, using a combination of algorithms and mathematics called "threshold logic" for mimicking the thought process (McCulloch & Pitts, 1943). Their model, known as MCP (stands for McCulloch and Pitts) neural model, is considered an ancestor of Artificial Neural Model. Following the great success of MCP, the concept of the perceptron was introduced by Rosenblatt (1958). His perceptron simulate a vision system and consists of four main units as shown in Figure 3.1; sensory unit which is the retina, projection unit where the impulses from the retina are transmitted to (this unit is often optional), association unit which receives the impulses from the projection area through random connections (this unit fires only if the sum of the impulses is equal or greater than the threshold), and response unit (works in the same way as projection and association unit).

Rosenblatt's perceptron, when the projection unit is omitted, is very similar to the modern perceptron which as shown in Figure 3.2 consists of: sensory units where the data is collected, association units where the linear and non-linear processes are performed and response units where the results from the association unit is passed. The association units add different weights to the data from the sensory units and apply a non-linear transformation (activation function) onto the threshold sum. The association units take all the inputs x from the sensory units and multiply them with different weights, then sum up the products and add bias b, this mathematically can be represented as:

$$f(y) = \sum_{p=1}^{n} (w_p \times x_p) + b^p \tag{3.1}$$

where p is the number of patterns, x_p is the p-th input (e.g. an image). The activation function f(y) amplifies/de-amplifies it and produces output y^p from the neuron. The non-linear activation function is the main difference between the early type and modern type of perceptrons. It was suggested by Bose & Liang (1996) that linear threshold (activation) function should be softened to simulate biological neural networks.

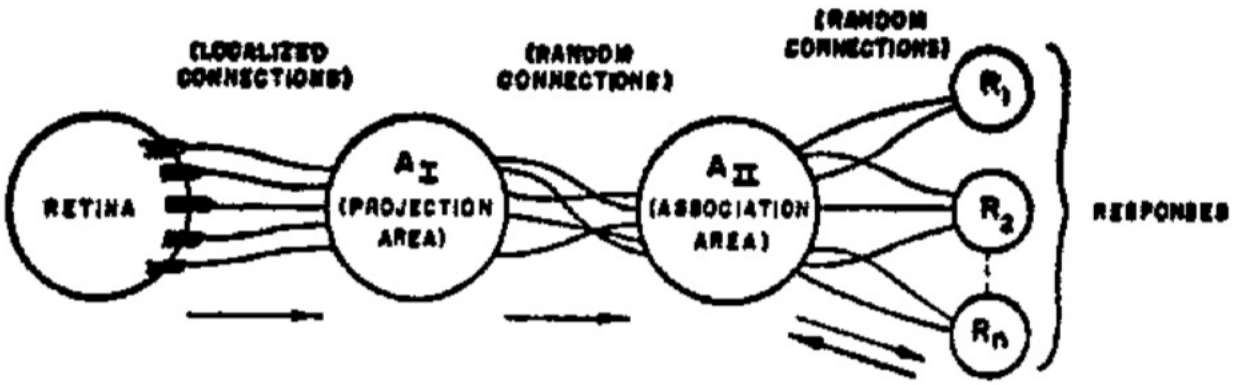


Figure 3.1: A schematic diagram of Rosenblatt's perceptron. Figure from Rosenblatt (1958)

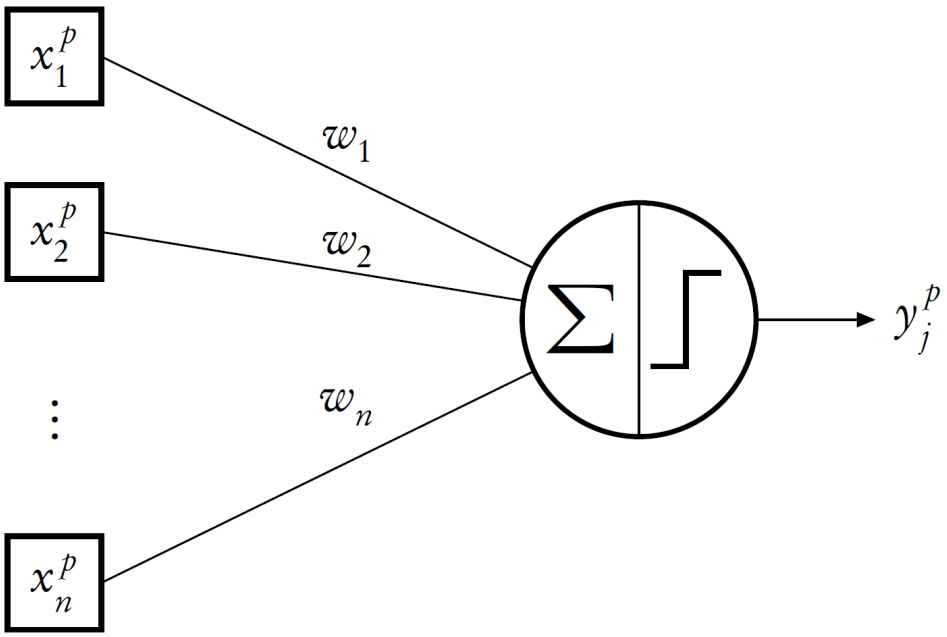


Figure 3.2: A schematic diagram of a modern perceptron, The single perceptron classifies input data, provided that the given problem is linearly-separable. The perceptron's output is a non-linear function of the input data, $\mathbf{x}$, projected onto the weight vector, $\mathbf{w}$. Figure from Bai et al. (2014)

3.1.1 Activation Function

The activation function or a non-linearity layer is a function that takes a vector and performs a fixed point-wise (non-linear) operation on it. Without the non-linearity layer, the network will only learn a linear combination of its input. We described here the four most commonly used activation functions.

Sigmoid

The *Sigmoid* activation function takes a real value and squashes it between 0 and 1 (same as $softmax$) (Bridle, 1990) as shown in Figure 3.3. This can be mathematically represented as:

$$f(x) = \frac{1}{(1 + exp^{-x})} \tag{3.2}$$

As with $softmax$, using the *sigmoid* function in the output layer as an activation function is very common practice. The main advantage of *sigmoid* is that it is continuous and differentiable and its derivative is relatively fast to compute (Goodfellow et al., 2016). Often when the neuron's activation saturates at either tail of 0 or 1, the back propogation algorithm fails at modifying its parameters because the gradient value at these regions is almost zero, which is called vanishing gradient. The same would then happen to the parameters of the preceding neural layers.

Hyperbolic Tangent

The $TanH$ activation function squashes a real number ($\in R$) to the range $\in [-1, 1]$ (Goodfellow et al., 2016) as shown in Figure 3.4, $TanH$ mathematically can be represented as:

$$f(x) = \frac{1 - exp^{-2x}}{1 + exp^{-2x}} \tag{3.3}$$

However, it has the same drawback as the sigmoid which tend to vanish gradient when the neuron's activation saturates at either tail of 0 or 1 (Goodfellow et al., 2016).

Rectified Linear Unit

The $ReLU$ activation function was introduced by Hahnioser et al. (2000) for a dynamical network, and used for the first time to train deep neural networks by Glorot et al. (2011). $ReLU$ sets negative values to 0 and returns only the positive values as shown in Figure 3.3. Mathematically can be represented as:

$$y_i = \begin{cases} x_i & \text{if} \quad x_i \geq 0 \\ 0 & \text{if} \quad x_i < 0 \end{cases} \tag{3.4}$$

ReLU is frequently used for hidden layers. Compared to other activation functions, *ReLU* has been proven to accelerate the convergence of stochastic gradient descent by a factor of 6 because of its linear non-saturating form (Krizhevsky et al., 2012; Nair & Hinton, 2010). Besides that, *ReLU* also has a simple (cheap) operation compared to the exponential in the *Sigmoid* case and gives better performance than $TanH$ (Glorot et al., 2011). Because of these advantages we used *ReLU* as an activation function for the output of the convolutional layers. However, *ReLU* removes all the negative values and only returns the positive ones, which might not be suitable with all Network's architectures and datasets.

Softmax

The $softmax$ activation function (Bridle, 1990), takes a vector of weight and produces a scalar output in the range between 0 and 1 (Convert scores $\in R$ to probabilities $\in [0, 1]$) as shown in Figure 3.4. $Softmax$ can be mathematically represented as:

$$f(x) = \frac{exp^{x_i}}{\sum_i exp^{x_i}} \tag{3.5}$$

where x is the output (vector of weight) from the previous layer. It is a common practice for $softmax$ to be used in the output layer of the network to predict the probability of each class, where the final output prediction is the class of the highest probability. Because of the failure of *sigmoid* and $TanH$ when the neuron's activation saturates at either tail of 0 or 1, we made use of $softmax$ in the output layer.

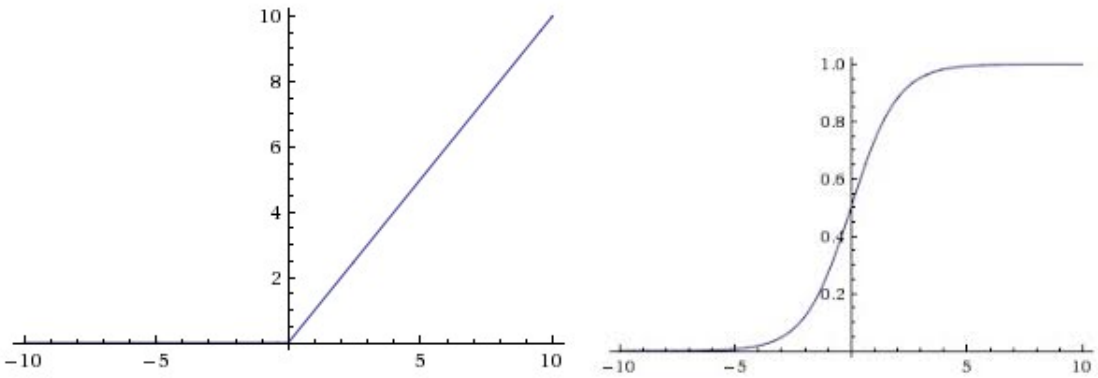


Figure 3.3: Left: Rectified Linear Unit (*ReLU*) activation function, which is zero when $x < 0$ and linear with slop 1 when $x > 0$. Right: *sigmoid* activation function squashes real number to range $\in [0, 1]$ (Staravoitau, 2018).

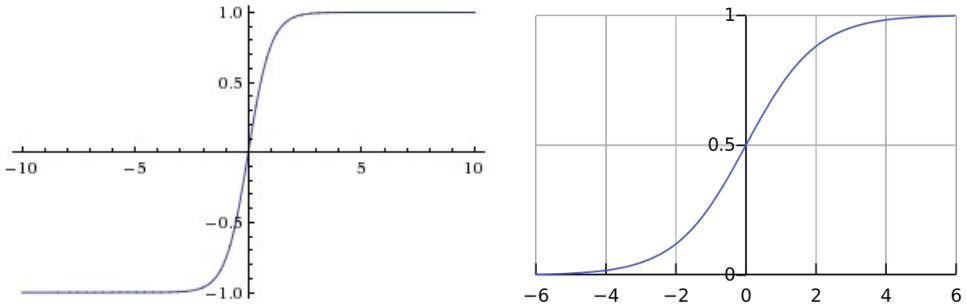


Figure 3.4: Left: $TanH$ activation function squashes real number to range $\in [-1, 1]$. Right: $softmax$ activation function squashes real number to range $\in [0, 1]$ (Staravoitau, 2018).

3.2 Multi-Layer Perceptron

A perceptron is fundamentally a linear function of input signal. Perceptrons can be placed side by side to construct a single one layer neural network. Different neural networks can be stacked to produce a multi-layer neural network, which is often known as multi-layer perceptrons (MLP) (Kawaguchi, 2000).

Artificial neural networks (ANN) (Cybenko, 1989) were the first implementation of MLP. Its computational model was inspired by the nervous system to approximate nonlinear functions from a set of inputs by combination of simple functions and gives an anticipated answer. ANN is capable of learning patterns in noisy and non-linear data which makes it a powerful approach for problems that are hard or impossible to solve by computers such as handwriting recognition or speech recognition. As shown in Figure 3.5, ANNs generally consist of interconnected neurons which may have many inputs and a single output like a biological neuron, where each input has assigned weight. Each input and its corresponding weight is then summed up and the result is fed into a non-linear activation function.

The neuron amplifies or de-amplifies the weights according to their relevance in minimizing the error rate and transforms the output, in order to update the weights continuously, this process is called Back propogation (BP). It was introduced by Henry J. Kelley (Kelley, 1960), which is a technique that can compute the backward propagation of errors. BP enables the optimization of DNNs and helps efficiently compute the gradient of parameters (weights) when a gradient descent algorithm is applied to learn parameters. A simpler version based only on the chain rule was developed by Dreyfus (1962). Due to the limitation of computers in terms of speed and computational ability at that time, back propogation was inefficient and not useful until the 1980s (Hecht-Nielsen, 1988), when Werbos (1982) published the first application of back propogation to neural networks for purposes of training ANNs. How BP is applied to DNN specially to Convolutional Neural Networks is explained in chapter 5.

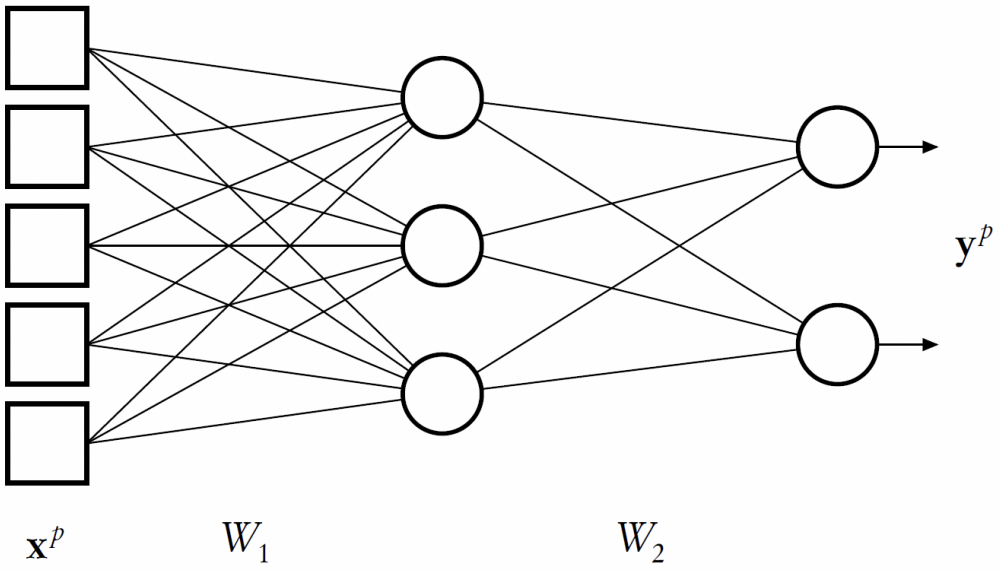


Figure 3.5: A collection of perceptrons combined to form a neural network. The network shown is a "deep neural network" because it has multiple hidden layers. Figure from Bai et al. (2014)

3.3 Deep Neural Networks (DNNs)

A DNNs is an ANN model with multiple hidden layers between the input and output layers, each subsequent hidden layers representing higher levels of abstraction of the input (Patil et al., 2017; Bengio, 2009). Information is passed through each layer, with the output of the previous layer providing input for the next layer. The first layer in a network is called the input layer, while the last is called an output layer. All the layers between the two are referred to as hidden layers. Each layer is typically a simple, uniform algorithm containing one kind of activation function (LeCun et al., 2015). DNNs make use of BP in order to tweak and adjust their internal parameters for better computation of the representation in each layer from the representation in the previous layer (Bai et al., 2014).

DNNs are typically networks, where the data flows from the input layer to the output layer without looping back as shown in Figure 3.6. This is often called Feedforward DNNs (Schmidhuber, 2015). An example of this type is Convolutional Neural Networks (ConvNet; Fukushima (1980)) which have wide applications in image and video recognition (LeCun et al., 1998), CNN is explained in some details in chapter 5. Another type of Neural Networks are Recurrent neural networks (RNNs; Gers & Schmidhuber (2001)), where the data can flow in any direction, RNNs are used widely for applications on sequential data such as language modeling and machine translation. Different DNNs models with different setups and architecture can be used for classification and regression applications to solve different problems.

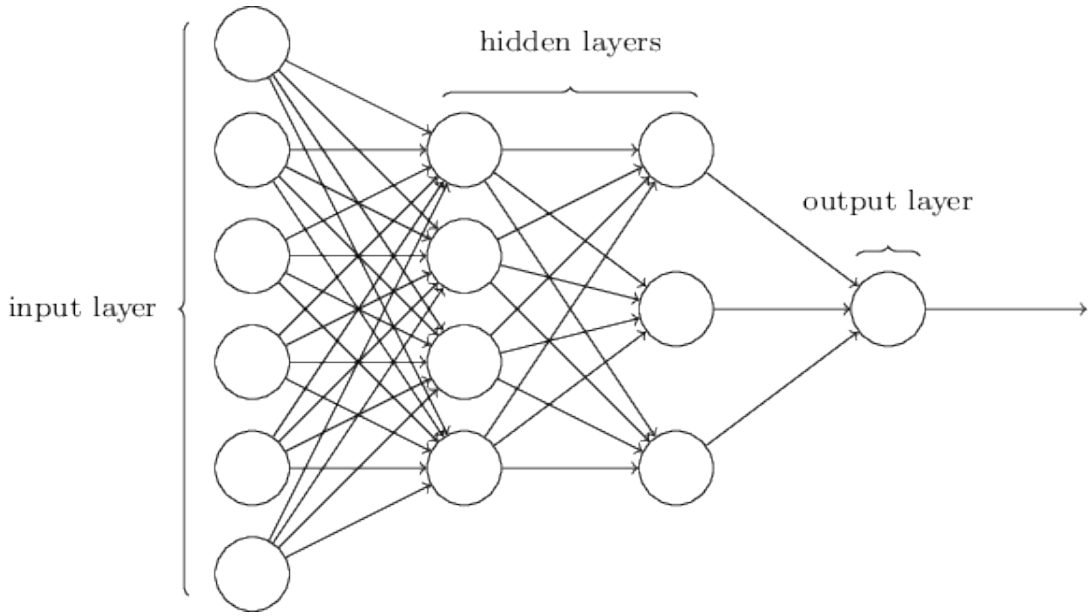


Figure 3.6: Feedforward neural network. Figure from Davim (2011)

3.4 Advantages and Limitations of DNNs

DNNs generally outperform other solutions in multiple domains like computer vision, easily can be applied to different fields and ability to handle high-dimensional big data. But is very expensive computationally, required a large amount of data and optimizing the hyper parameters is not straightforward process.

3.4.1 Advantages

- Can learn and model very complex non-linear relationships versus traditional machine learning algorithms learning and outperforms other solutions in multiple domains like computer vision, speech analysis, natural language processing and games.
- Reduces the need for feature engineering, which is a very time-consuming part of machine learning practice, specially with deluge data.
- Can be readily applied to many fields. Generally, DNN architecture can be adapted to new problems relatively easily (e.g. Vision, time series, language etc) using techniques like convolutional neural networks, recurrent neural networks, etc.
- Ability to handle high-dimensional data (i.e. images, videos).

3.4.2 Limitations

- Requires a large amount of data.

- Computationally very expensive to train. The most complex models take weeks to train using hundreds of machines equipped with expensive GPUs.
- Optimizing the DNN (training methods, hyper-parameters, etc.) is not easy to determine and can only be done through experiments.
- What is learned is not easy to understand versus traditional machine learning algorithms (e.g. decision trees, etc).

3.5 Tools and Libraries

The number of available open-source libraries for Deep Learning is constantly growing. TensorFlow (Abadi et al., 2015), Caffe (Jia et al., 2014), Theano (Al-Rfou et al., 2016) and Torch (Collobert et al., 2002) are the four major libraries when it comes to interface, optimization, training speed, scalability and readability of the code.

TensorFlow (Abadi et al., 2015) is an open source software library for numerical computation developed by the Google Brain Team within Google's Machine Intelligence research organization. AsTheano Tensorflow has a central idea of data flow graphs. TensorFlow can be used from C, C++ and Python programs. TensorFlow also allows a graphical visualization ("TensorBoard" tool) and to scope on sub-levels of computations for debugging purposes. We developed our CNN model based on the **Keras** Deep Learning framework (Chollet et al., 2015) using the **TensorFlow** backend. The choice was made mainly because of its consistent and simple APIs, its minimization of the number of user actions required for common use cases, and it also provides clear feedback (debugging methods) upon user error.

Chapter 4

Convolutional Neural Networks

Computer Vision (CV) is an active research field with very broad applications in many disciplines. Its main concern is how computers can be used to gain high-level understanding from digital images or videos (Ballard & Brown, 1982). CV includes collecting, processing, analysing and extracting of features from real world data in order to produce numerical or symbolic information. Pattern recognition is considered a main task in CV, which addresses the problem of classifying input data represented as vectors, into categories (Vijendra et al., 2011).

The application of Convolutional Neural Networks (CNNs) to computer vision goes back to 1998, achieving good results for handwritten digit classification (LeCun et al., 1998). With the development of computing technology, CNNs have recently shown state-of-the-art performance on image classification (Krizhevsky et al., 2012). Important improvements have been achieved in visual recognition of many categories (Jiang & Learned-Miller, 2016).

4.1 Description

The Convolutional Neural Networks (ConvNet; Fukushima (1980)) is a type of feed-forward neural network model for deep learning. CNNs consists of multilayer structure, namely convolutional layers (which act as a filter for the input images to extract features), followed by an activation non-linearity function such as *tanh*, *sigmoid* or *ReLU*, then pooling layers, which are vector to scalar transformations that operate on local regions of an image to generate a representative value of the pixels in that region. Average pooling computes the average of pixels in a region, while max pooling uses the value of the pixel with the highest intensity in the region, optionally followed by fully connected layers (Agarap, 2017). Generally the architecture of CNNs is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal) (Goodfellow et al., 2016). An example of CNN model is shown in Figure 4.1.

In our case, the input to the convolutional layer is an image of size $m \times m \times c$ where m is the

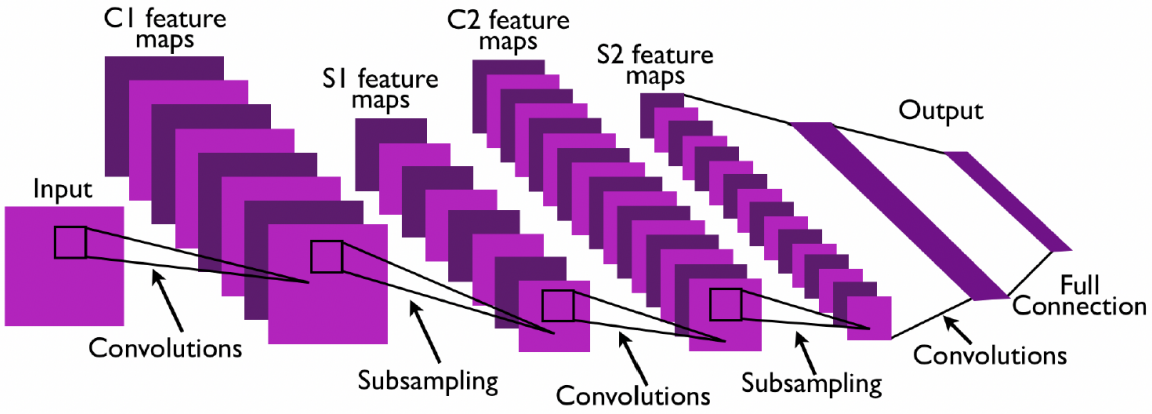


Figure 4.1: A Convolutional Neural Networks example with tow hidden layers. Figure from LeCun et al. (2010)

width and height of the image and c is number of bands (e.g. a gray-scale image has 1 band and an RGB image has 3 bands), size $n \times n \times g$ (where $n < m$ and $g \leq c$) and initial values of filters (weight matrices) is user-defined. Each weight element is convolved with the image to produce k channels called feature maps of size $m \times n + 1$. Each feature map will be sub-sampled with a pooling layer (either max, min or average) over contiguous regions (usually 2×2 for small images or 3×3 for larger one) (Krizhevsky et al., 2012).

4.2 CNN Layers

4.2.1 Convolutional Layer

Convolutional (linear) layers are essentially made up of neurons that receive inputs. Each neuron is connected locally to its inputs from the previous layer. The inputs are each assigned a random weight, and a dot product is performed. The scalar output is then passed through a non-linear activation function. The output k-th feature map Z of a single neuron is a result of a non-linear transformation; and can be mathematically represented as:

$$y^k = w^k * x = f(\sum_{i'=1}^{n} \sum_{j'=1}^{n} w^k_{i',j'} x_{i+i',j+j'} + b^k) \tag{4.1}$$

$$Z^k = f(y^k) \tag{4.2}$$

Where x is the activations of the input neurons connected to the neurons (i, j) in the following convolutional layer, w is an $n \times n$ weight matrix of the convolutional filter, b is the bias, $*$ is the convolution operator and $f()$ is a non-linear activation function (usually sigmoid or softmax). In

general, convolutional layer can be thought of as a function which applies a linear transformation on a vectorial input of dimension n and outputs a vector of dimension k. The whole network describes the non-linear mapping between raw image pixels and their class scores.

4.2.2 Feature Pooling Layer

The pooling layer is a vector to scalar transformation that operates on local regions of an image to generate a representative value of the pixels in that region (Graham et al., 2014). The main purpose of this layer is to reduce the resolution of the feature map from the preceding layer, through eliminating position information of the features, in order to achieve spatial invariance (not affected by the position of the objects in the image). The size of the pooling window can be arbitrary, and windows can overlap depending on the user choice (Scherer et al., 2010). The main two types of pooling used are max and average Figure 4.2, which both produce a feature map of lower resolution. Max pooling uses the value of the pixel with the highest intensity in the region, via applying a window function $w(x, y)$ of size$(n \times n)$ to the input($N \times N$ feature map), and computes the maximum in the neighbourhood. Mathematically the Max pooling function can be represented as:

$$y_j = Max_{N \times N}(y_i^{n \times n} w(n, n)) \tag{4.3}$$

Average pooling computes the average values of pixels in a region, multiplies it by a trainable scalar B, adds a trainable bias b and passes the result through the non-linearity function f:

$$y_j = f(B \sum_{N \times N} y_i^{n \times n} + b) \tag{4.4}$$

Also, it allows activation functions such as TanH and Sigmoid to not get stuck in the saturation mode (e.g. gradient equal to 0).

4.2.3 Fully Connected (FC) Layer

After several convolutional and pooling layers, finally the FC layers comes, where the high-level reasoning in the neural network is done. Neurons in a fully connected layer have connections to all activations in the previous layer, as seen in regular neural networks. Their activations can hence be computed with a matrix multiplication followed by a bias offset (Ma & Lu, 2017). FC classifies the input image into various classes and can be replaced by any traditional classifier like Support Vector Machine (SVM) and Random Forest classifiers.

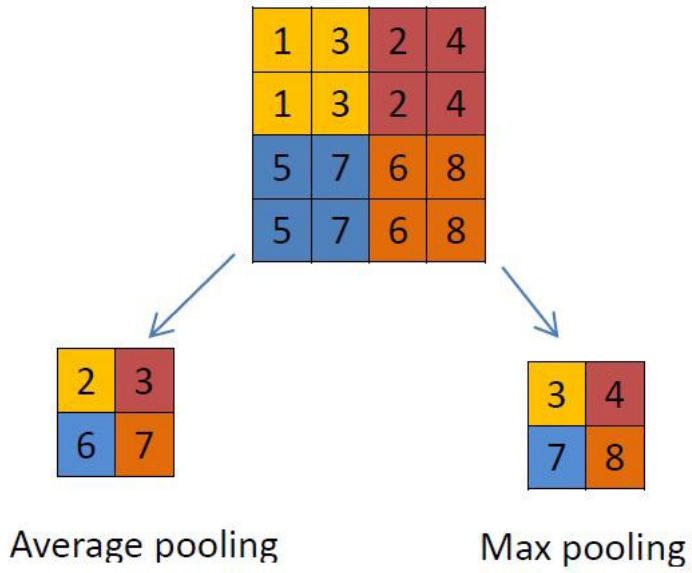


Figure 4.2: Max and Average pooling with a $2times2$ filter example of a single depth slice of size 4×4 input. https://goo.gl/xohkdV

4.3 Training a Convolutional Network

The first step in training a CNN model is defining the batch size and number of epochs. Afterwards, the model will randomly initialize and assign weights across all the node connections and predict the output. The next step is the BP, where the model will assess and improve the accuracy of the output automatically by adjusting the weights across all the node connections. This process is repeated until the training is finished. These can generally be summarized as :

1- Take some examples from the training dataset, usually a batch of 16–256 elements.

2- Calculate error of the network for this batch.

3- Adjust the weights of the network to reduce this error, backwards, layer-by-layer.

4- Repeat 1–3 until the training is finished.

4.3.1 Batch Size

Batch size is number of observations (images) the model will select from the entire dataset to process each time. This can be any number between 1 and several hundred. For the classification of images, batch size values are usually a power of two that fit nicely into the memory requirements of the GPU or CPU hardware. Generally, small batch sizes will be more computationally

weighty since the process will take longer time especially with very large datasets, while large batch sizes will results in lower feedback signal.

4.3.2 Epochs

Number of epochs represents the number of times the model sees the entire dataset, often called iteration. Each epoch during the training, the model learns new features and adjust its parameters to improve the accuracy of predictions (Goodfellow et al., 2016). In our training dataset, we have 13,780 images. Having a batch size of 128 will require 107 passes for one epoch to be completed. In general, the more complex the features and relationships in the dataset, the more epochs will be required for the model to learn, adjust the weights, and minimize the error in predictions.

4.3.3 Learning Rate

The learning rate controls the update of the weight during the training from epoch to epoch. In general, choosing a very small rate will results in slower learning process while choosing a very high learning rate will results in high fluctuations and unstable Learning process. Choosing an optimum learning rate for any problem is usually done via experiments.

4.3.4 Weight Initialization

Keras framework provides different kinds of weight initializers, for example, RandomNormal and RandomUniform initializers, which generate tensors with a normal and a uniform distribution receptively. Also, Zeros and One initializers, generate tensors initialized to 0 and 1 receptively. In our model, we initialised the kernel of each layer with RandomNormal initializer.

4.3.5 Loss Function

The loss (often called objective) function quantifies the capacity of the network by comparing the ground truth to the prediction of the model and measuring the error (loss) (Goodfellow et al., 2016). Loss function takes weights, biases, and samples from the training set as an inputs. We used a multi-categorical Cross Entropy function as a loss function which is the most popular and commonly used in image classification. Cross Entropy can be mathematically represented as :

$$L(x, y) = -\sum_{j} y_j \times log(\frac{exp(x_i)}{\sum_j exp(x_j)}) + (1 - y_i) \times log(\frac{1}{1 + exp(x_j)}) \tag{4.5}$$

Where L is a loss function, x is a vector of n predictions, and y is a vector vector full of 0's besides a 1 in the corresponding class dimension.

4.3.6 Back Propogation Algorithm for CNNs

The BP is the process of adjusting the model's weights across all the node connections to minimize the error (loss) in the output predictions and reach the local minima (minimum error) (LeCun et al., 2015) . BP computes backwards the derivative (gradient) of the weights with respect to the loss function (output of the model). Once these gradients have been computed, it is straightforward to compute the gradients with respect to the weights of each module.

In detail, for each batch, the model computes the prediction and its associated loss E, then the BP algorithm propagates the error in order to compute the partial derivatives $\frac{\partial E}{\partial Z}$ (Naumov, 2017). The BP equation is applied repeatedly to propagate gradients through all the model Figure 4.3, starting from the output at the top (where the network produces its prediction) all the way to the bottom (where the data is fed) until a sufficiently small error is achieved.

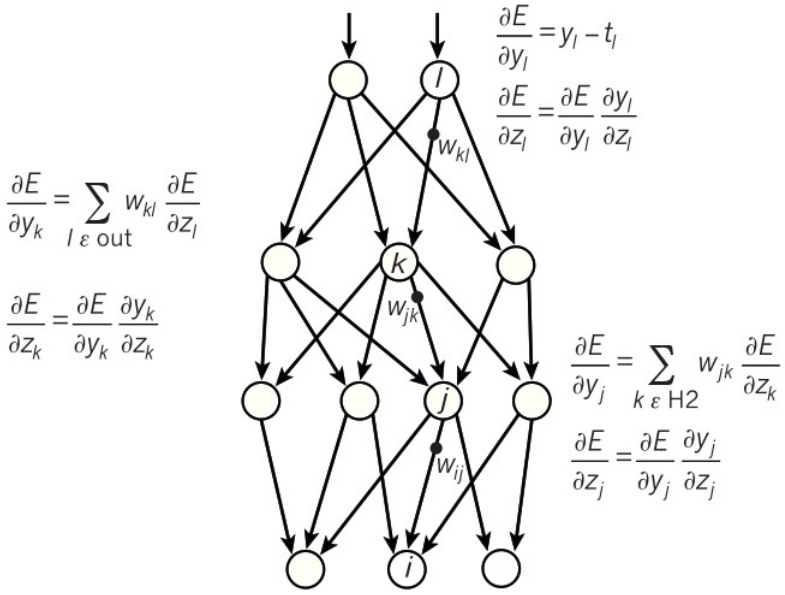


Figure 4.3: Compare the output with the ground truth to compute the error derivative. Figure from (LeCun et al., 2015)

4.3.7 Optimization Algorithms

Optimization algorithms are algorithms used to minimize the loss function by updating the parameters of the model in the opposite direction of the gradient of the loss function in order for the model to reach the minimum error (local minimum). Stochastic Gradient Descent (SGD) is the most popular algorithms for optimizing loss functions, especially in image classification problem, which we made use of in our model.

Stochastic Gradient Descent (SGD)

SGD algorithm (Ruder, 2016) is used in the real-time on-line processing , where the parameters (weights) are adjusted in each epoch for each training sample, then the accuracy average and the error of all the samples in the epoch will be computed. SGD performs a parameter update for each training example x_i and label y_i as follow:

$$\theta_{t+1} = \theta_t - \lambda \cdot \delta_{\theta t} L(f_{\theta t}(x_i), y_i) \tag{4.6}$$

SGD is relatively faster than other similar optimising algorithms, but it performs frequent updates with a high variance that results in high fluctuation in the objective function.

4.3.8 Regularization

Generally Deep Neural Networks, including CNNs, usually suffers from overfitting (LeCun et al., 2015). This phenomenon happens when the accuracy of the training dataset, during the training of the model, converges towards perfection while decreasing on the validation and test datasets. This is mainly because the model is relying on the presence of the same features each time on the training dataset instead of learning new features, which will cause it to fail on predicting unseen datasets. To overcome this problem, regularization techniques were suggested which help the DNNs model to avoid overfitting such as data augmentation and dropout and early stopping. We made use of all the below three techniques for the training of our model.

Data augmentation

Data augmentation is a combination of methods of increasing the size of the training set when the size of the dataset is not big enough to train the model, as in our case, to prevent the model from memorizing all of it. This can take several forms depending of the dataset and the problem we trying to solve. For example, in image classification problem, if the objects are supposed to be invariant to rotation, such as galaxies in our case, applying different kind of rotations and flipping to the original images will increase the size of the dataset significantly.

Dropout

Dropout function (Hinton et al., 2012), is one of the most recent regularisation techniques that has shown great success especially in Deep neural nets(large number of parameters), Because it is difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. *Dropout* is a technique for addressing this problem, which avoids over-fitting by setting randomly the output of the previous layer's neurons to 0, to force the model to learn robust features during the training instead of depending on the presence of the same features

each time (Srivastava et al., 2014). During the testing, Dropout acts like a form of model averaging over all different levels of the model.

Early Stopping

Early stopping is one of the regularization techniques that acts like a mentor of the training process. Its main task is to watch the performance measure of the model, and stop the training before the model begins to overfit on the training dataset, or once the model starts to throw "Nan" values. In practice, it is very common technique used to train CNNs. The early stopping technique was used in the training, with patience = 7, which means to wait for 7 epochs before stopping the training in case the performance is getting worst. We also used TerminateOnNan built-in function in *keras* to terminate the training immediately if $loss = nan$.

4.4 Advantages, Disadvantages and Achievements

CNNs has proved to have a great potential in solving vision tasks, an example is Optical Character Recognition in Google maps and street view (Wojna et al., 2017). CNNs Mimic the human visual system in extracting the image's relevant features automatically regardless the positions. In contrast, training a CNNs is computationally expensive, required a lot of data and tuning its hyper-parameters is not straightforward process.

Advantages

- Implicitly extracts relevant features from the image, same as the human visual system.
- Achieved a great degree of shift and distortion invariance.
- Automatic feature extraction for the given task, which greatly saves human time and more closely resembles true artificial intelligence.
- Performs universally well with various topologies and is not very sensitive to pre-training normalization.
- Relatively easier to train than MLPs.

Disadvantages

- Requires a large amount of data.
- Computationally very expensive to train. Complex models with large number of layers take a long time (several weeks) to train even when fast machines equipped with expensive GPUs are used.

- Optimizing the hyper-parameters of CNNs is not easy to determine and can only be done through experiments.
- What is learned is not easy to understand versus traditional machine learning algorithms (e.g. decision trees, logistic regression, etc).

Achievements

CNNs has proved to have a great potential in solving vision tasks. This potential has attracted a large number of researchers aiming to solve vision tasks which they achieved with great results, for example:

- Optical Character Recognition in Google maps and street view (Wojna et al., 2017).
- Traffic sign detection (Escalera et al., 2011).
- Alexnet in imageNet competition (Krizhevsky et al., 2012).
- Very Deep Convolutional Networks for Large-Scale Image Recognition which they provided VGG model in blind competition of exploring different architectures (Simonyan & Zisserman, 2014).

Chapter 5

Experiments

In this chapter we show the detailed specifications of the architecture for each model and the hyper parameters we used to train it. We constructed three different models with different numbers of convolutional layers (1,2 and 3 convolutional layers) and different filters and depth sizes, starting off with shallower model and then go deeper by adding more layers and changing the number of filters. We ran three different experiments and found that, dropout, learning rate and batch size of 0.5, 0.0001 and 128 respectively work better to solve our problem leaving all the other parameters as default. Although these values for the hyper-parameters lead to an excellent performance, we did not attempt a fully-fledged optimization procedure. Different methods for optimizing the hyper-parameters will be investigated in detail in future work with a larger and less augmented dataset.

5.1 Networks Architecture

The content of radio images at our disposal can be described by a small number of features then the number of pixels, because the number of pixels that contains information after the cleaning process is limited, or in other word, most of the pixels values of the image after removing the background noise becomes zero. Based on this, we expected simple network architecture models (i.e. a small number of convolutional layers) for the classification of the compact and extended radio sources. We constructed three different models with different numbers of convolutional layers (1,2 and 3 convolutional layers) and different filters and depth sizes, starting off with shallower model and then go deeper by adding more layer and changing the number of filters.

5.1.1 Model 1

We built model 1 using only one convolutional layer with 32 filters of size 8×8 to filter the gray input image $150 \times 150 \times 1$. A ReLU activation function was applied to the output before

it is sub-sampled using a max pooling function with kernel of size 4×4, followed by a Fully-Connected (FC) layer with 32 channels and a *ReLU* activation function then a *Dropout* was added, Followed by another FC layer with only 4 channels. A $softmax$ activation function was applied to the final output to calculate the probability for each class. The architecture of the model and its learned parameters are summarized in Table 5.1.

Layer Type	Function	Depth	kernel Size	Parameters
Convolutional	Convolution	32	8×8	2080
Pooling	Max pooling	32	4×4	
FC1	Fully Connected Layer	32		1401888
Dropout	Dropout(0.5)	32		
FC2	Fully Connected Layer	4		132
Total number of learned parameter				1,404,100

Table 5.1: Model 1 summary

5.1.2 Model 2

The architecture of model 2 consists of two convolutional layers. The first convolutional layer filters the $150 \times 150 \times 1$ input image (gray image) with 32 filters of size 5×5, its output was sub-sampled with a Max-pooling function with 3×3 kernel size. The second convolutional layer has 64 filters with size of 5×5, followed by a Max-pooling function of size 3×3. The last Max-pooling was followed by 2 FC layers. The first one had 64 channels with *ReLU* activation function and a *Dropout*. The second FC layer has 4 channels which performs the classification, where its output was fed to the $softmax$ activation function to calculate the probability for each class. The architecture of the model and its learned parameters are summarized in Table 5.2.

Layer Type	Function	Depth	kernel Size	Parameters
Convolutional	Convolution	32	5×5	832
Pooling	Max pooling	32	3×3	
Convolutional	Convolution	64	5×5	51264
Pooling	Max pooling	64	3×3	
FC1	Fully Connected Layer	64		1048640
Dropout	Dropout(0.5)	64		
FC2	Fully Connected Layer	4		260
Total number of learned parameter				1,100,996

Table 5.2: Model 2 summary

5.1.3 Model 3

As shown in Figure 5.1, the architecture of model 3 consists of five trainable layers, three convolutional layers and two FC layers. The first convolutional layer filters the $150 \times 150 \times 1$

input image (gray image) with 32 filters of size 3×3, The second Convolutional layer filters the output of the first one with 64 filters of size 3×3 and the last convolutional layer filters the output of the second layer with 194 filters of size 3×3. A *ReLU* activation function was applied to the output of the all three convolutional layers. The output of all the convolutional layer were sub-sampled with Max-pooling with of size 2×2, the last Max-pooling was followed by 2 FC layers. The first one had 194 channels with *ReLU* activation function and a *Dropout*. The second FC layer performs the classification, where its output was fed to the $softmax$ activation function to calculate the probability for each class. The architecture of the model and its learned parameters are summarized in Table 5.3.

Layer Type	Function	Depth	kernel Size	Parameters
Convolutional	Convolution	32	3×3	320
Pooling	Max pooling	32	2×2	
Convolutional	Convolution	64	3×3	18496
Pooling	Max pooling	64	2×2	
Convolutional	Convolution	194	3×3	111938
Pooling	Max pooling	194	2×2	
FC1	Fully Connected Layer	194		12194258
Dropout	Dropout(0.5)	194		
FC2	Fully Connected Layer	4		780
Total number of learned parameter				12,325,792

Table 5.3: Model 3 summary

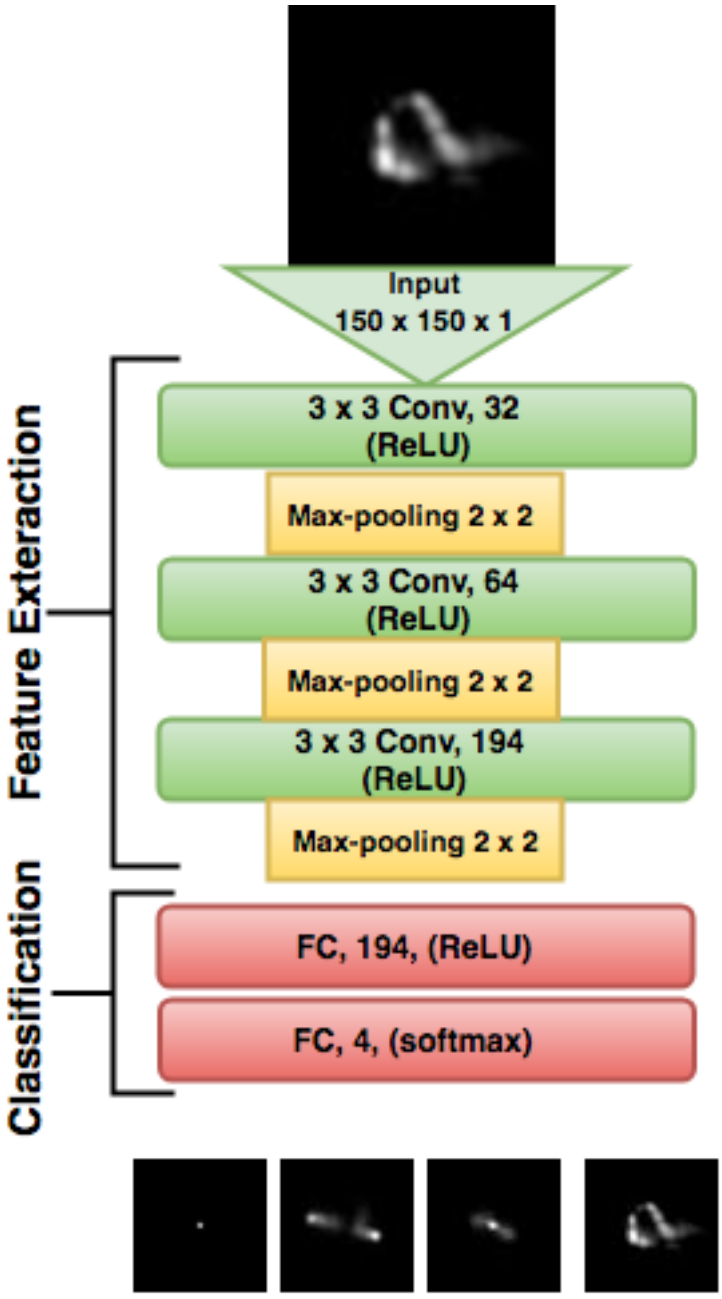


Figure 5.1: Deep CNN architecture of Model 3. In this Figure each convolutional layer (Conv) is followed by ReLU activation function and Max-pooling function. The number in each box represents the number of channels in the corresponding feature map.

5.2 Hyperparameters

As mentioned in chapter 3, one of the disadvantage of DNNs is the optimization of the hyper-parameters, which can only be done through experiments, which is expensive computationally. There are large number of parameters that can be optimized, some of them specify the structure of the network of the model itself (layers, filters, type of pooling layers, etc.), and the others determine how the network is trained, like the type of the optimizers and its parameters (learning rate, etc.), the regularization techniques(dropout, data augmentation, etc.) and the type of loss function (Goodfellow et al., 2016).

To simplify the process, and also because we are only applying the CNN as a tool to solve our astrophysical problem here, we will only focus on three parameters, batch size (BS), dropout (DO) and learning rate (LR) and leave the others parameters that we didn't discuss in the previous chapter to their default values (e.g momentum, clip, decay, etc.). Our main concern will be about the general performance of each model, which will be evaluated through the learning curves (loss and accuracy).

We conducted three different experiments for each of our three models. First we chose initial values for each parameter, then we evaluate the performance using the learning curves and increase values of the parameters gradually to get better performance. From the method we used, it might not be very clear which parameters has the most impact on the results, but because all we need here is a good model that can classify our radio sources correctly we have adopted this method to optimize the parameters.

For the number of epochs, we were very careful not to train our model for so many epochs, due to our highly augmented dataset (12,780 from only 837), assuming that, it might lead to over-fitting. Considering this, a decision was made to train each of our models for only 50 epochs, to avoid any possibility of over-fitting. The results show that, 50 epochs was enough to achieve a very good results but we believe still not the best we can get. Future work will focus on classifying large number of sources manually (at least 5000 for each class), to be used as a training dataset, and retrain our models for bigger numbers of epochs to achieve better and more robust results.

In order to optimize the hyper-parameters (BS, DO and LR), we first chose one initial values for each, and then increased them gradually to see the difference in performance to find the most optimum values to solve our problem.

We trained each of our three models three times using different values of DO, BS and LR. To train our models, we use the SGD algorithm (section 4.3.7) for the parameters updates, categorical cross-entropy as a loss function(section 4.3.5) for the training loss, The weights are initialized using Random Normal initializer (section 4.3.4). In this work we do not use any pre-trained model. Every time the network is trained all weights are reinitialized.

For the DO, Park and Kwak (Park & Kwak, 2017) found that, dropout=0.1 work better.

We used 0.1 as an initial value to start and then increased it to 0.3 and 0.5. For the LR and BS, we chose our initial values from the *Keras* documentation [*]. In the *keras* example, LR in the SDG optimiser is 0.01, BS is 32 . We also increased the LR and BS gradually in our second and third experiment to be 0.001, 64 and 0.0001, 128 respectively.

Figure 5.2, 5.3, 5.4, 5.5, 5.6, and 5.7 show the performance of the different values on the training and validation phases of our three models, where we can see, the initial values has the worst performance, specially in model 2 Figure 5.4, where the accuracy has a fixed value of ≈ 0.15 during the training and validation phases, and a loss of ≈ 12 in both phases as shown in Figure 5.5. The second and third set of values has different accuracy and loss values during the first epochs of training and a relatively similar performance after the first 20 epochs. From the Figures, the third set of values , which shown in Table 5.4, have clearly more accurate and smooth results.

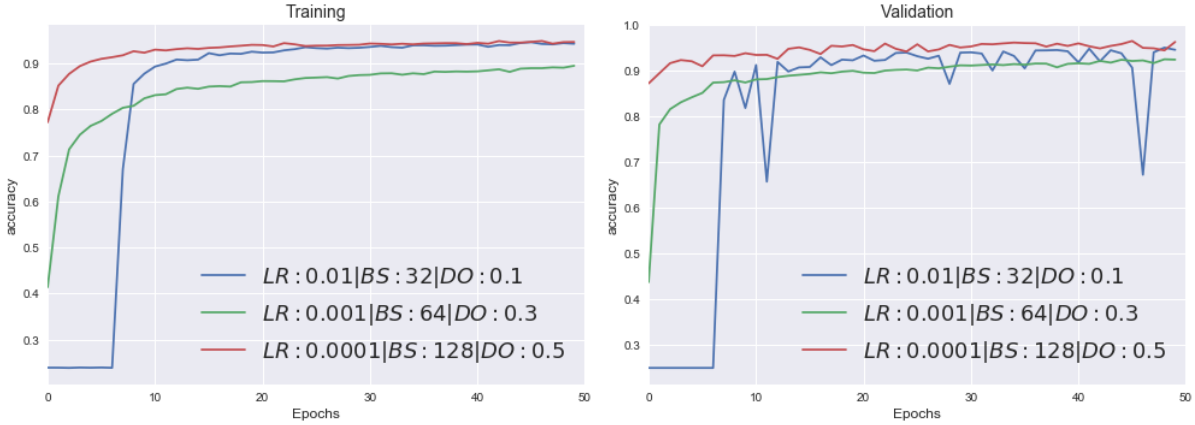


Figure 5.2: Model 1 hyper-parameters optimisation

Hyper-parameters	Values
Batch Size	128
Dropout Rate	0.5
Learning Rate	0.0001
Epochs	50

Table 5.4: Hyper-parameters used to train the CNN model

[*] https://keras.io/

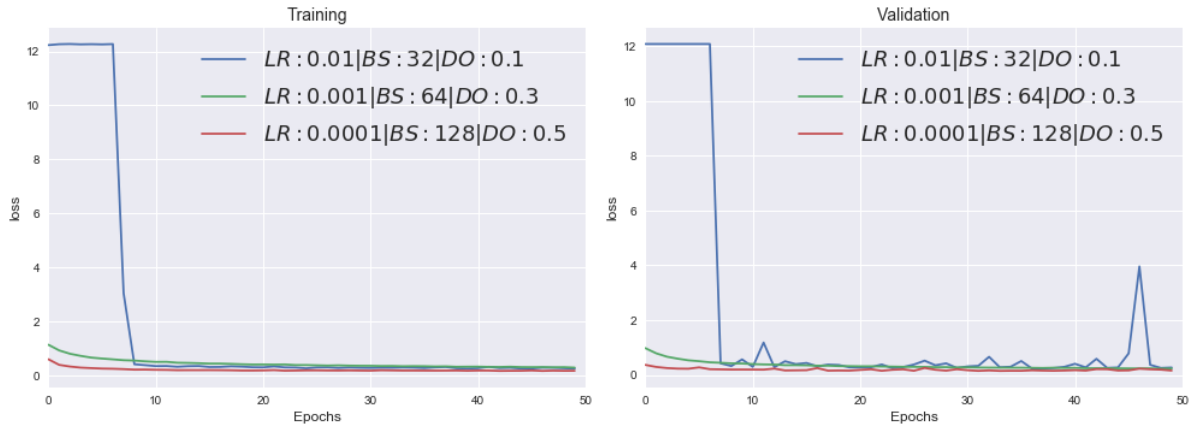


Figure 5.3: Model 1 hyper-parameters optimisation

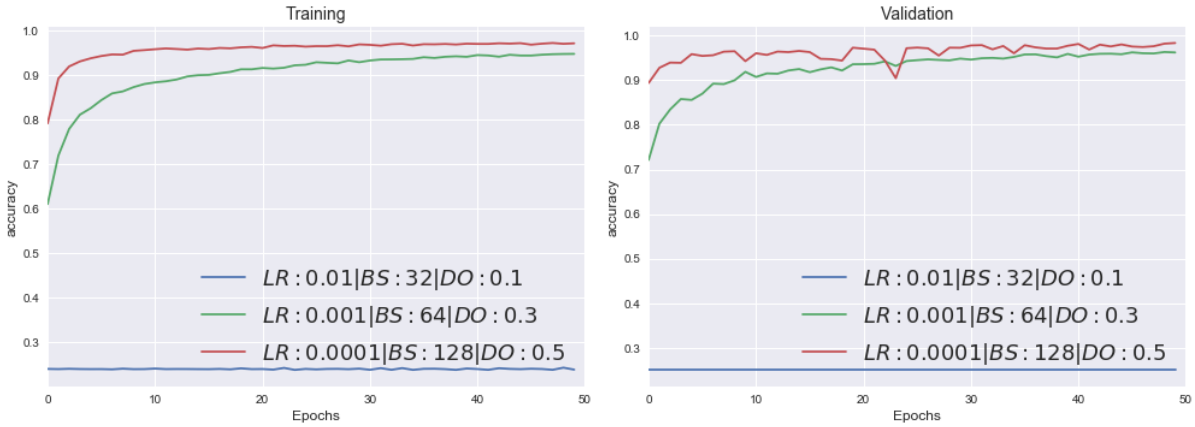


Figure 5.4: Model 2 hyper-parameters optimisation

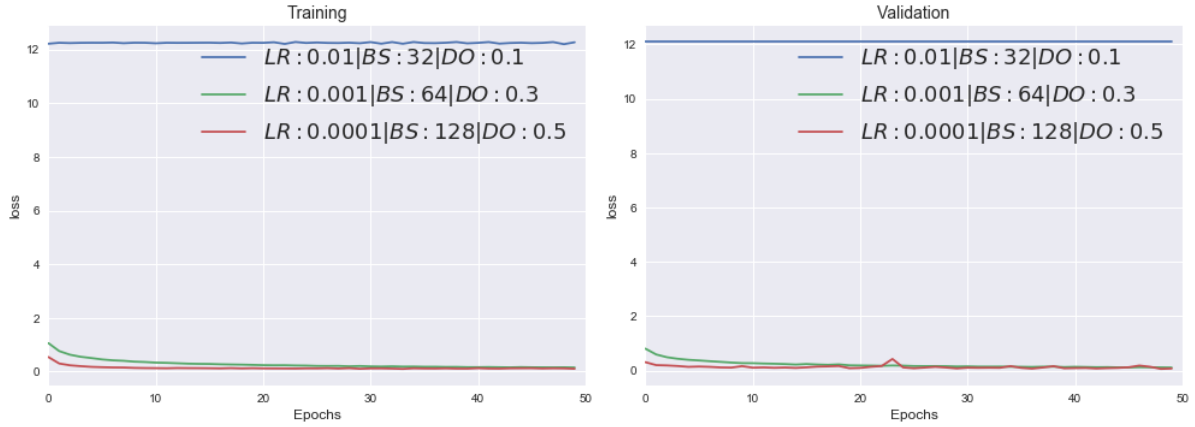


Figure 5.5: Model 2 hyper-parameters optimisation

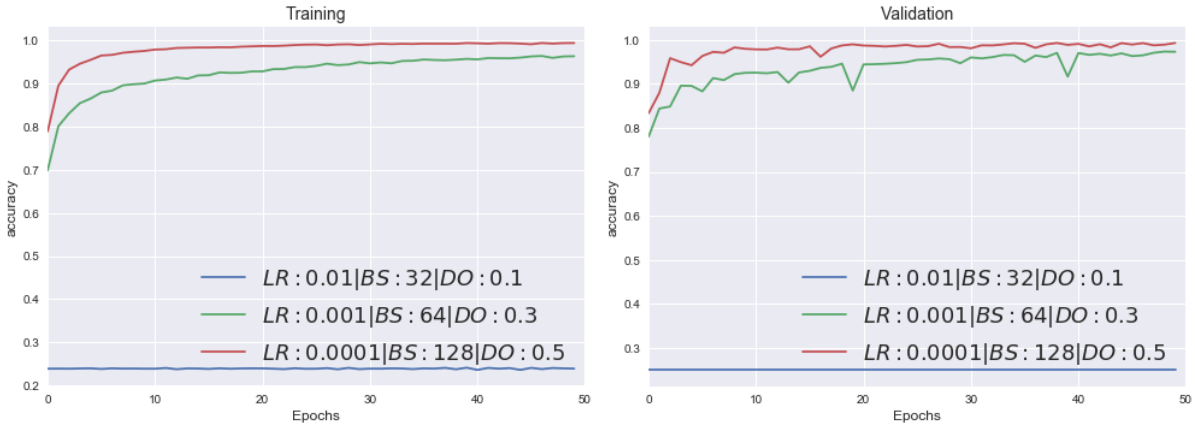


Figure 5.6: Model 3 hyper-parameters optimisation

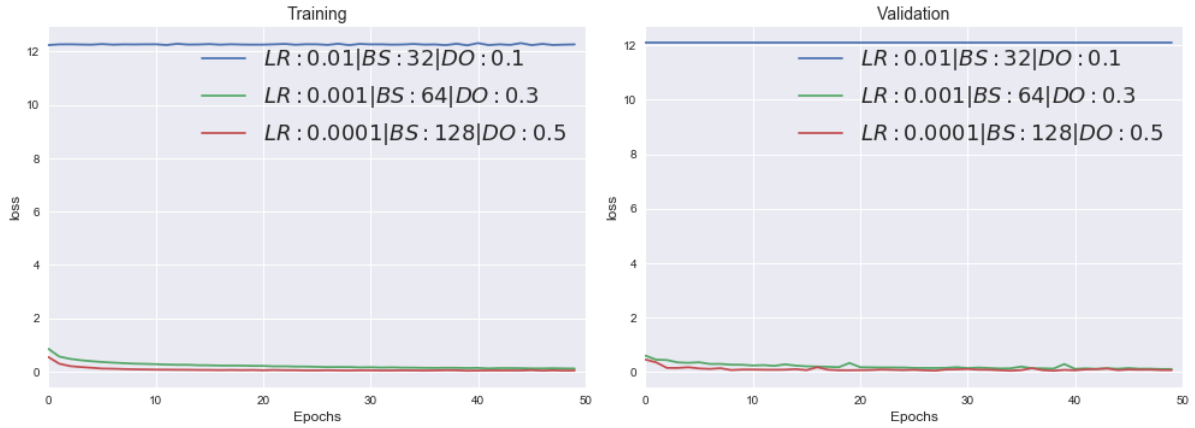


Figure 5.7: Model 3 hyper-parameters optimisation

Chapter 6

Results

We have trained each of our three models for 20 times. Classification reports, confusion matrices and learning curves were generated for each. Model 1, Model 2 and Model 3 achieved an average accuracy, precision, recall and F1 of 0.93, 0.97 and 0.98 respectively on the testing data set. We concluded that, quantitatively and qualitatively, Model 3 outperforms Model 1 and 2 in all aspects.

6.1 Metrics and Quantitative Experimental Results

For measuring the performance of the deep learning model, the learning curve is generally used, which provides a quantitative measure of the performance of the model in terms of accuracy (how accurate the prediction is by comparing the ground truth with the prediction) and loss (defined in 4.3.5), with respect to the number of epochs on the training and validation dataset (Urry & Sollich, 2012). Losses and accuracy curves for the training and validation datasets were plotted with respect to the number of epochs for each experiment. Must be noted that, rather than while spotting the accuracy and the loss values from the learning curve plots at the end of the training and validation phases, we used the evaluate *keras* built-in function to produce them, which is more accurate.

In order to assess how accurately our models are able to predict the different morphology classes, the precision (P), recall (R) and F1 scores (Fawcett, 2006) were calculated using our test dataset, based on the number of true positive (TP), false positive (FP) and false negative (FN) classifications as given below:

$$recall = \frac{TP}{TP + FN} \tag{6.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6.3}$$

where:

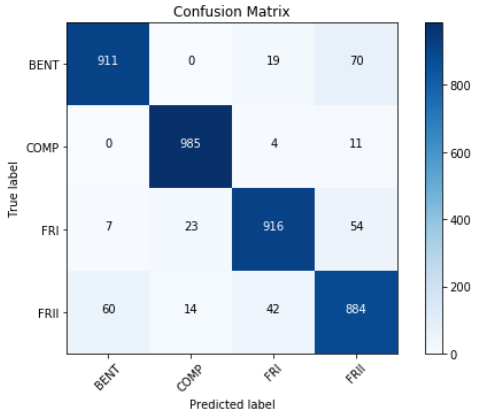
- True positive (TP) is when the source for instance is predicted as FRI and it is actually FRI.
- False positive (FP) is when the source is predicted for instance as FRI and it is actually not FRI.
- False negative (FN) is when the source is predicted for instance as not FRI and it is actually FRI.

Generally, the precision is the best measure to check for over-fitting, while the recall provides a quantitative measure of the correctly classified sources. Recall and precision are also called completeness and reliability (or sensitivity) respectively (Hopkins et al., 2015). The higher value of recall and precision the better performance of the model (Ivezic et al., 2014). F1 score is a joint metric of the recall and the precision, which is a good measure of the classification performance. Higher value of F1 score means better performance of the classification. For each model, the classification report is provided, where the precision, recall and F1 score were calculated for each of our 4 classes.

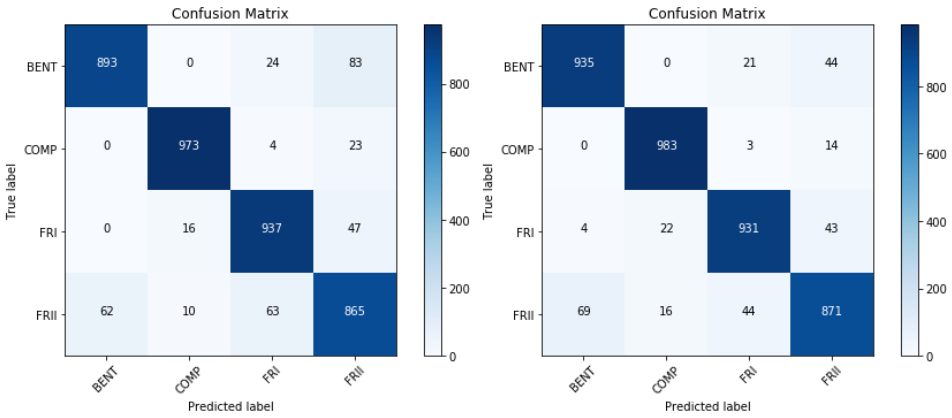
The Confusion Matrix (CM) is used to visualize the performance of Machine Learning algorithms Stehman (1997). Its stability (not affected by the imbalance number of images for the different classes) was proved generally in the case of multi-class classification tasks Machart & Ralaivola (2012). A CM is essentially a matrix where each row represents the instances in a predicted class ($Xaxis$) while each column represents the instances in an actual class ($Yaxis$), the classes ("actual" and "predicted") on both axis are identical which enables comparison with the ground truth, and makes it easier to interpret and see if the system confuses the classes. A confusion matrix of our testing dataset for each model was generated, where the recall, TP, FP and FN values were plotted for compact, FRI, FRII and BENT sources.

6.1.1 Model 1 Results

Model 1 achieved an average accuracy of 0.9325, and an average precision, recall and F1 of 0.9263, 0.9257 and 0.9256 with a standard deviation (std) of 0.0062, 0.0059 and 0.0057 respectively on the testing data set. The classification reports Figure 6.2 show the precision, recall, F1 score and the number of sources for each of our 4 classes. Three main groups of values for precision, recall and F1 score of 0.91, 0.92 and 0.93 were observed. Because the results in each group are very similar, we chose one model from each group as a representative. Model 1_1, 1_3 and 1_9 from the 0.91, 0.92 and 0.93 group respectively. To check for the performance during the training and validation phases, the accuracy and loss curves for model 1_1, 1_3 and 1_9 are shown in Figure 6.3 and 6.4. As a quantitative check, the confusion matrices were produced for model 1_1, 1_3 and 1_9 in Figure 6.1, where the exact number of the truly classified and miss-classified compact, FRI, FRII and BENT sources in the testing dataset was presented.



(a)



(b) (c)

Figure 6.1: The Confusion Matrix of Model 1 on our testing dataset, where the true labels at Y axis and Predicted labels at the X axis, the blue boxes at the diagonal represents the recall values. (a): Model 1_1, (b): Model 1_3 and (c): Model1_9.

Model 1_1

	precision	recall	f1-score	support
BENT	0.9417	0.9040	0.9224	1000
COMP	0.9631	0.9920	0.9773	1000
FRI	0.9308	0.9280	0.9294	1000
FRII	0.8756	0.8870	0.8813	1000
avg / total	0.9278	0.9277	0.9276	4000

Model 1_2

	precision	recall	f1-score	support
BENT	0.9315	0.9110	0.9211	1000
COMP	0.9638	0.9850	0.9743	1000
FRI	0.9337	0.9160	0.9248	1000
FRII	0.8675	0.8840	0.8757	1000
avg / total	0.9241	0.9240	0.9240	4000

Model 1_3

	precision	recall	f1-score	support
BENT	0.9351	0.8930	0.9136	1000
COMP	0.9740	0.9730	0.9735	1000
FRI	0.9115	0.9370	0.9241	1000
FRII	0.8497	0.8650	0.8573	1000
avg / total	0.9176	0.9170	0.9171	4000

Model 1_4

	precision	recall	f1-score	support
BENT	0.9276	0.9350	0.9313	1000
COMP	0.9628	0.9830	0.9728	1000
FRI	0.9319	0.9310	0.9315	1000
FRII	0.8961	0.8710	0.8834	1000
avg / total	0.9296	0.9300	0.9297	4000

Model 1_5

	precision	recall	f1-score	support
BENT	0.9292	0.9190	0.9241	1000
COMP	0.9548	0.9930	0.9735	1000
FRI	0.9248	0.9230	0.9239	1000
FRII	0.8880	0.8640	0.8758	1000
avg / total	0.9242	0.9247	0.9243	4000

Model 1_6

	precision	recall	f1-score	support
BENT	0.9369	0.9210	0.9289	1000
COMP	0.9609	0.9830	0.9718	1000
FRI	0.9200	0.9310	0.9254	1000
FRII	0.8910	0.8750	0.8829	1000
avg / total	0.9272	0.9275	0.9273	4000

Model 1_7

	precision	recall	f1-score	support
BENT	0.9343	0.8960	0.9148	1000
COMP	0.9593	0.9900	0.9744	1000
FRI	0.9200	0.9200	0.9200	1000
FRII	0.8652	0.8730	0.8691	1000
avg / total	0.9197	0.9197	0.9196	4000

Model 1_8

	precision	recall	f1-score	support
BENT	0.9260	0.9130	0.9194	1000
COMP	0.9467	0.9950	0.9703	1000
FRI	0.9335	0.9130	0.9232	1000
FRII	0.8812	0.8680	0.8746	1000
avg / total	0.9219	0.9223	0.9219	4000

Model 1_9

	precision	recall	f1-score	support
BENT	0.9625	0.9240	0.9429	1000
COMP	0.9601	0.9860	0.9729	1000
FRI	0.9402	0.9280	0.9341	1000
FRII	0.8908	0.9140	0.9023	1000
avg / total	0.9384	0.9380	0.9380	4000

Model 1_10

	precision	recall	f1-score	support
BENT	0.9651	0.9400	0.9524	1000
COMP	0.9507	0.9740	0.9622	1000
FRI	0.9204	0.9270	0.9237	1000
FRII	0.9166	0.9120	0.9143	1000
avg / total	0.9382	0.9283	0.9281	4000

Model 1_11

	precision	recall	f1-score	support
BENT	0.9417	0.9040	0.9224	1000
COMP	0.9631	0.9920	0.9773	1000
FRI	0.9308	0.9280	0.9294	1000
FRII	0.8756	0.8870	0.8813	1000
avg / total	0.9278	0.9277	0.9276	4000

Model 1_12

	precision	recall	f1-score	support
BENT	0.9315	0.9110	0.9211	1000
COMP	0.9638	0.9850	0.9743	1000
FRI	0.9337	0.9160	0.9248	1000
FRII	0.8675	0.8840	0.8757	1000
avg / total	0.9241	0.9240	0.9240	4000

Model 1_13

	precision	recall	f1-score	support
BENT	0.9292	0.9190	0.9241	1000
COMP	0.9548	0.9930	0.9735	1000
FRI	0.9248	0.9230	0.9239	1000
FRII	0.8880	0.8640	0.8758	1000
avg / total	0.9242	0.9247	0.9243	4000

Model 1_14

	precision	recall	f1-score	support
BENT	0.9369	0.9210	0.9289	1000
COMP	0.9609	0.9830	0.9718	1000
FRI	0.9200	0.9310	0.9254	1000
FRII	0.8910	0.8750	0.8829	1000
avg / total	0.9272	0.9275	0.9273	4000

Model 1_15

	precision	recall	f1-score	support
BENT	0.9343	0.8960	0.9148	1000
COMP	0.9593	0.9900	0.9744	1000
FRI	0.9200	0.9200	0.9200	1000
FRII	0.8652	0.8730	0.8691	1000
avg / total	0.9197	0.9197	0.9196	4000

Model 1_16

	precision	recall	f1-score	support
BENT	0.9551	0.9400	0.9424	1000
COMP	0.9607	0.9740	0.9722	1000
FRI	0.9304	0.9370	0.9337	1000
FRII	0.9066	0.9020	0.9043	1000
avg / total	0.9382	0.9383	0.9381	4000

Model 1_17

	precision	recall	f1-score	support
BENT	0.9443	0.9060	0.9248	1000
COMP	0.9693	0.9800	0.9844	1000
FRI	0.9300	0.9400	0.9300	1000
FRII	0.8752	0.8830	0.8791	1000
avg / total	0.9297	0.9297	0.9296	4000

Model 1_18

	precision	recall	f1-score	support
BENT	0.9451	0.8830	0.9236	1000
COMP	0.9640	0.9830	0.9635	1000
FRI	0.9215	0.9270	0.9141	1000
FRII	0.8397	0.8750	0.8673	1000
avg / total	0.9176	0.9170	0.9171	4000

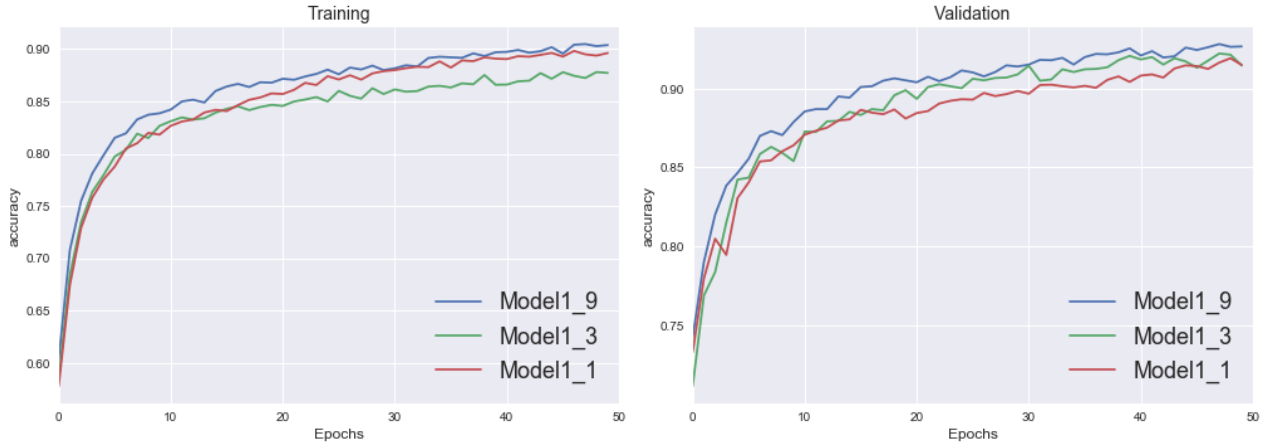
Model 1_19

	precision	recall	f1-score	support
BENT	0.9525	0.9140	0.9329	1000
COMP	0.9501	0.9760	0.9629	1000
FRI	0.9302	0.9180	0.9241	1000
FRII	0.8808	0.9040	0.8923	1000
avg / total	0.9284	0.9280	0.9280	4000

Model 1_20

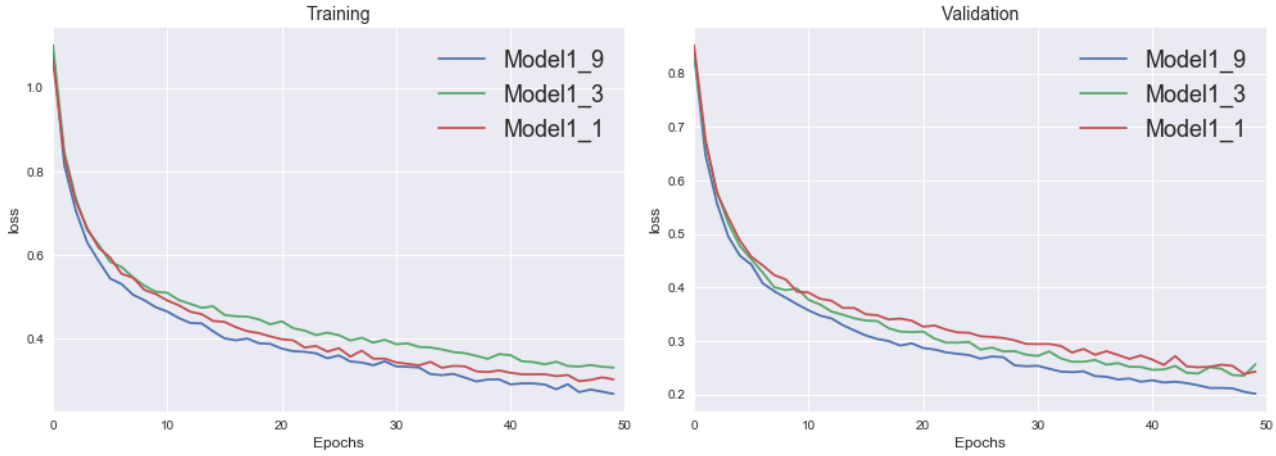
	precision	recall	f1-score	support
BENT	0.9443	0.8960	0.9148	1000
COMP	0.9493	0.9800	0.9744	1000
FRI	0.9300	0.9300	0.9200	1000
FRII	0.8552	0.8630	0.8691	1000
avg / total	0.9197	0.9172	0.9196	4000

Figure 6.2: Model 1 classification reports.



(a) (b)

Figure 6.3: Learning curves of Model 1 (Model 1_1, 1_3 and 1_9) with respect to the number of epochs in the training and validation phase. (a): Training accuracy curves, (b): Testing accuracy curves.



(a) (b)

Figure 6.4: Learning curves of Model 1 (Model 1_1, 1_3 and 1_9) with respect to the number of epochs in the training and validation phase. (a): Training loss curves, (b): Testing loss curves.

6.1.2 Model 2 Results

Model 2 achieved an average accuracy of 0.97, and an average precision, recall and F1 of 0.9665, 0.9664 and 0.9661 with STD of 0.0069, 0.0070 and 0.0070 respectively on the testing data set. The classification reports of the 20 trained model are shown in Figure 6.6. Three main groups of values for precision, recall and F1 score of 0.95, 0.96 and 0.97 were achieved. Model 2_3, 2_4 and 2_8 were chosen as a representative for 0.95, 0.96 and 0.97 groups respectively. To check for the performance during the training and validation, The accuracy and loss curves for model 2_3, 2_4 and 2_8 are shown in Figure 6.7 and 6.8, where no sign of over-fitting was spotted. The confusion matrices for model 2_3, 2_4 and 2_8 are shown in Figure 6.1.

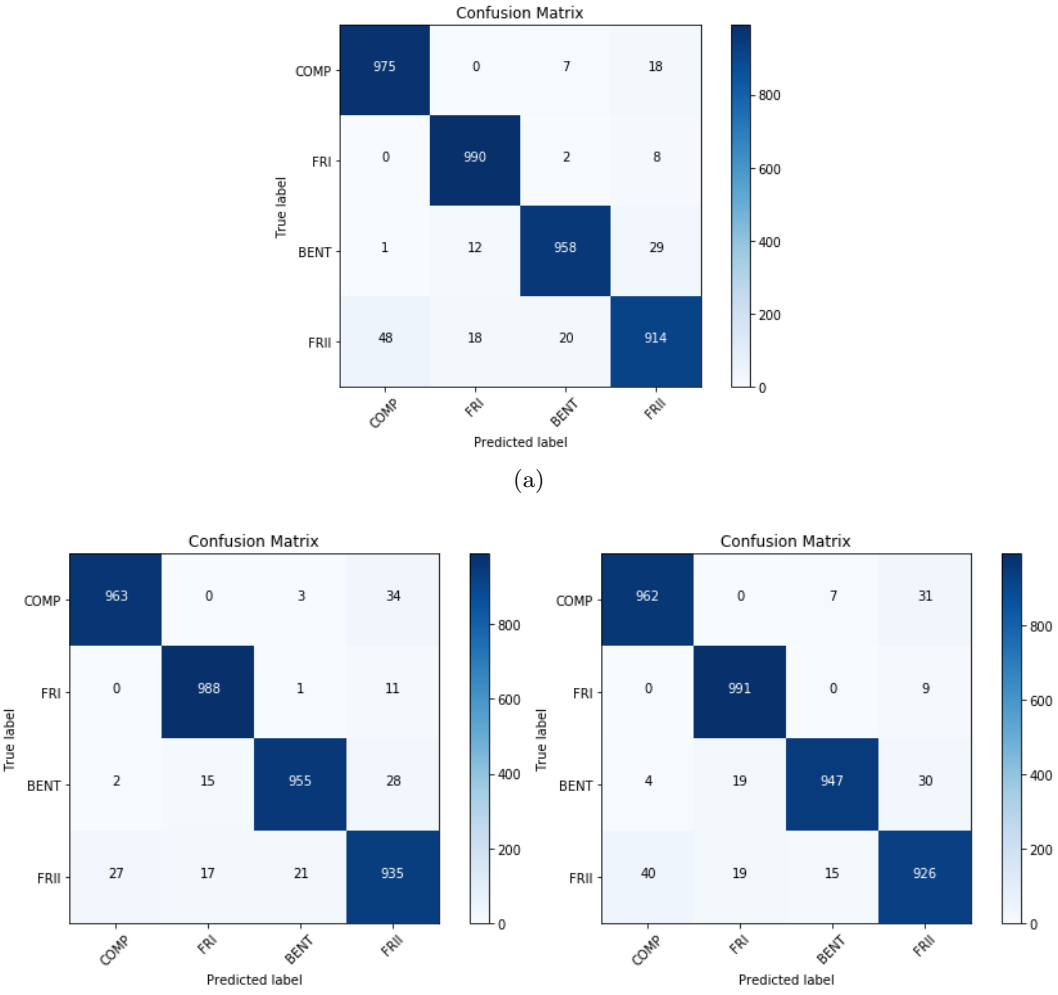


Figure 6.5: The Confusion Matrix of Model 2 on our testing dataset, where the true labels at Y axis and Predicted labels at the X axis, the blue boxes at the diagonal represents the recall values. (a): Model 2_3, (b): Model 2_4 and (c): Model2_8

Model 2_1

	precision	recall	f1-score	support
COMP	0.9521	0.9750	0.9634	1000
FRI	0.9706	0.9900	0.9802	1000
BENT	0.9706	0.9580	0.9643	1000
FRII	0.9432	0.9140	0.9284	1000
avg / total	0.9591	0.9593	0.9591	4000

Model 2_2

	precision	recall	f1-score	support
COMP	0.9708	0.9630	0.9669	1000
FRI	0.9686	0.9880	0.9782	1000
BENT	0.9745	0.9550	0.9646	1000
FRII	0.9276	0.9350	0.9313	1000
avg / total	0.9604	0.9603	0.9603	4000

Model 2_3

	precision	recall	f1-score	support
COMP	0.9688	0.9640	0.9664	1000
FRI	0.9674	0.9780	0.9727	1000
BENT	0.9754	0.9510	0.9630	1000
FRII	0.9195	0.9370	0.9282	1000
avg / total	0.9578	0.9575	0.9576	4000

Model 2_4

	precision	recall	f1-score	support
COMP	0.9705	0.9550	0.9627	1000
FRI	0.9660	0.9930	0.9793	1000
BENT	0.9766	0.9620	0.9693	1000
FRII	0.9352	0.9380	0.9366	1000
avg / total	0.9621	0.9620	0.9620	4000

Model 2_5

	precision	recall	f1-score	support
COMP	0.9563	0.9620	0.9591	1000
FRI	0.9631	0.9910	0.9768	1000
BENT	0.9773	0.9470	0.9619	1000
FRII	0.9297	0.9260	0.9279	1000
avg / total	0.9566	0.9565	0.9564	4000

Model 2_6

	precision	recall	f1-score	support
COMP	0.9567	0.9720	0.9643	1000
FRI	0.9760	0.9740	0.9750	1000
BENT	0.9766	0.9610	0.9688	1000
FRII	0.9281	0.9300	0.9291	1000
avg / total	0.9594	0.9593	0.9593	4000

Model 2_7

	precision	recall	f1-score	support
COMP	0.9709	0.9660	0.9684	1000
FRI	0.9781	0.9840	0.9811	1000
BENT	0.9818	0.9690	0.9753	1000
FRII	0.9348	0.9460	0.9404	1000
avg / total	0.9664	0.9663	0.9663	4000

Model 2_8

	precision	recall	f1-score	support
COMP	0.9686	0.9870	0.9777	1000
FRI	0.9821	0.9870	0.9845	1000
BENT	0.9760	0.9770	0.9765	1000
FRII	0.9641	0.9400	0.9519	1000
avg / total	0.9727	0.9728	0.9727	4000

Model 2_9

	precision	recall	f1-score	support
COMP	0.9799	0.9760	0.9780	1000
FRI	0.9765	0.9960	0.9861	1000
BENT	0.9859	0.9780	0.9819	1000
FRII	0.9607	0.9530	0.9568	1000
avg / total	0.9757	0.9758	0.9757	4000

Model 2_10

	precision	recall	f1-score	support
COMP	0.9715	0.9530	0.9621	1000
FRI	0.9677	0.9890	0.9782	1000
BENT	0.9627	0.9560	0.9594	1000
FRII	0.9293	0.9330	0.9311	1000
avg / total	0.9578	0.9577	0.9577	4000

Model 2_11

	precision	recall	f1-score	support
COMP	0.9613	0.9690	0.9651	1000
FRI	0.9733	0.9860	0.9796	1000
BENT	0.9699	0.9670	0.9685	1000
FRII	0.9430	0.9260	0.9344	1000
avg / total	0.9619	0.9620	0.9619	4000

Model 2_12

	precision	recall	f1-score	support
COMP	0.9644	0.9740	0.9692	1000
FRI	0.9744	0.9890	0.9816	1000
BENT	0.9770	0.9770	0.9770	1000
FRII	0.9538	0.9300	0.9418	1000
avg / total	0.9674	0.9675	0.9674	4000

Model 2_13

	precision	recall	f1-score	support
COMP	0.9798	0.9680	0.9738	1000
FRI	0.9794	0.9980	0.9886	1000
BENT	0.9791	0.9850	0.9821	1000
FRII	0.9625	0.9500	0.9562	1000
avg / total	0.9752	0.9752	0.9752	4000

Model 2_14

	precision	recall	f1-score	support
COMP	0.9664	0.9810	0.9736	1000
FRI	0.9694	0.9870	0.9781	1000
BENT	0.9711	0.9780	0.9746	1000
FRII	0.9733	0.9340	0.9533	1000
avg / total	0.9700	0.9700	0.9699	4000

Model 2_15

	precision	recall	f1-score	support
COMP	0.9839	0.9760	0.9799	1000
FRI	0.9794	0.9970	0.9881	1000
BENT	0.9744	0.9880	0.9811	1000
FRII	0.9764	0.9530	0.9646	1000
avg / total	0.9785	0.9785	0.9784	4000

Model 2_16

	precision	recall	f1-score	support
COMP	0.9673	0.9810	0.9741	1000
FRI	0.9780	0.9770	0.9775	1000
BENT	0.9770	0.9770	0.9770	1000
FRII	0.9637	0.9510	0.9573	1000
avg / total	0.9715	0.9715	0.9715	4000

Model 2_17

	precision	recall	f1-score	support
COMP	0.9740	0.9770	0.9755	1000
FRI	0.9732	0.9860	0.9796	1000
BENT	0.9810	0.9780	0.9795	1000
FRII	0.9677	0.9550	0.9613	1000
avg / total	0.9740	0.9740	0.9740	4000

Model 2_18

	precision	recall	f1-score	support
COMP	0.9679	0.9600	0.9639	1000
FRI	0.9701	0.9780	0.9740	1000
BENT	0.9700	0.9660	0.9680	1000
FRII	0.9501	0.9540	0.9521	1000
avg / total	0.9645	0.9645	0.9645	4000

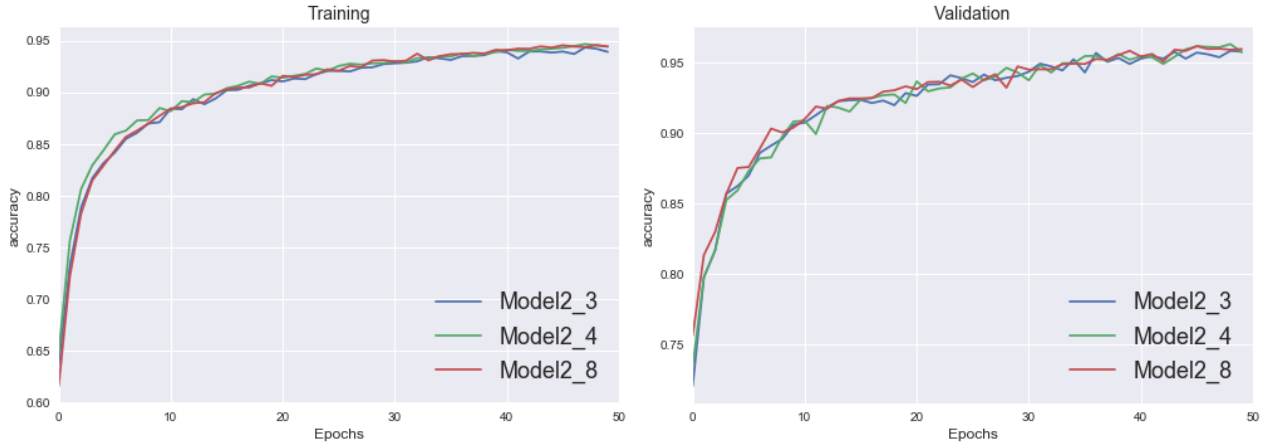
Model 2_19

	precision	recall	f1-score	support
COMP	0.9567	0.9720	0.9643	1000
FRI	0.9760	0.9740	0.9750	1000
BENT	0.9766	0.9610	0.9688	1000
FRII	0.9281	0.9300	0.9291	1000
avg / total	0.9594	0.9593	0.9593	4000

Model 2_20

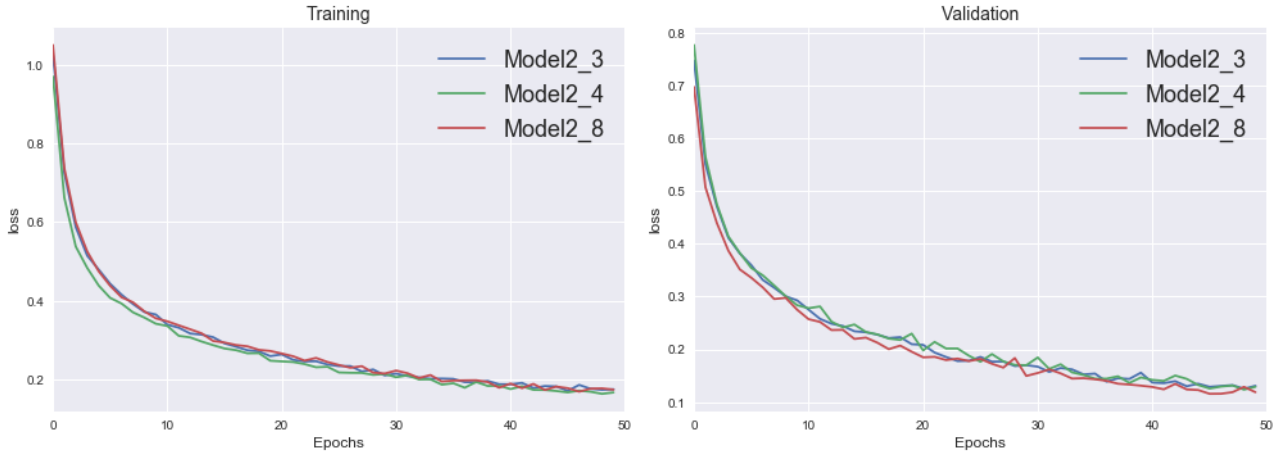
	precision	recall	f1-score	support
COMP	0.9664	0.9780	0.9722	1000
FRI	0.9755	0.9960	0.9857	1000
BENT	0.9849	0.9810	0.9830	1000
FRII	0.9650	0.9370	0.9508	1000
avg / total	0.9730	0.9730	0.9729	4000

Figure 6.6: Model 2 classification reports.



(a) (b)

Figure 6.7: Learning curves of Model 2 (Model 2_3, 2_4 and 2_8) with respect to the number of epochs in the training and validation phase. (a): Training accuracy curves, (b): Testing accuracy curves.



(a) (b)

Figure 6.8: Learning curves of Model 1 (Model 2_3, 2_4 and 2_8) with respect to the number of epochs in the training and validation phase. (a): Training loss curves, (b): Testing loss curves.

6.1.3 Model 3 Results

Model 3 achieved an average accuracy of 0.98, and an average precision, recall and F1 of 0.9837, 0.9836 and 0.9836 with STD of 0.0101, 0.0102 and 0.0102 respectively on the testing data set. The classification reports of the 20 trained model are shown in Figure 6.10. Three groups of values for precision, recall and F1 score of 0.97, 0.98 and 0.99 were achieved. Model 3_1, 3_7 and 3_14 were chosen as a representative for 0.97, 0.98 and 0.99 groups respectively.The accuracy and loss curves for model 2_3, 2_4 and 2_8 are shown in Figure 6.11 and 6.12, where no sign of over-fitting was spotted. The confusion matrices for model 3_1, 3_7 and 3_14 are shown in Figure 6.1.

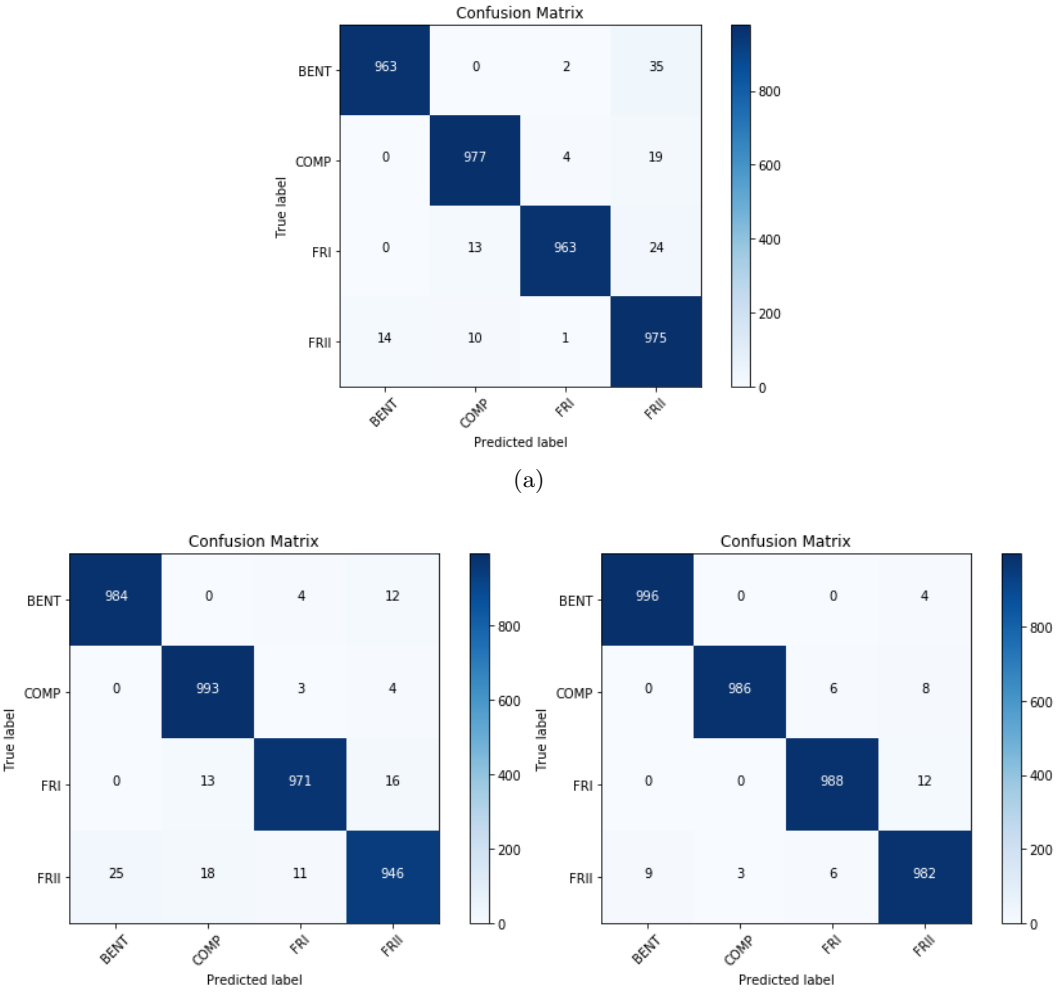


Figure 6.9: The Confusion Matrix of Model 2 on our testing dataset, where the true labels at Y axis and Predicted labels at the X axis, the blue boxes at the diagonal represents the recall values. (a): Model 3_1, (b): Model 3_7 and (c): Model3_14

Model 3_1

	precision	recall	f1-score	support
BENT	0.9857	0.9630	0.9742	1000
COMP	0.9770	0.9770	0.9770	1000
FRI	0.9928	0.9630	0.9777	1000
FRII	0.9259	0.9750	0.9498	1000
avg / total	0.9703	0.9695	0.9697	4000

Model 3_2

	precision	recall	f1-score	support
BENT	0.9800	0.9810	0.9805	1000
COMP	0.9817	0.9650	0.9733	1000
FRI	0.9847	0.9680	0.9763	1000
FRII	0.9361	0.9670	0.9513	1000
avg / total	0.9706	0.9702	0.9703	4000

Model 3_3

	precision	recall	f1-score	support
BENT	0.9595	0.9950	0.9769	1000
COMP	0.9820	0.9830	0.9825	1000
FRI	0.9771	0.9830	0.9801	1000
FRII	0.9749	0.9320	0.9530	1000
avg / total	0.9734	0.9732	0.9731	4000

Model 3_4

	precision	recall	f1-score	support
BENT	0.9772	0.9840	0.9806	1000
COMP	0.9704	0.9850	0.9777	1000
FRI	0.9888	0.9670	0.9778	1000
FRII	0.9550	0.9550	0.9550	1000
avg / total	0.9728	0.9728	0.9727	4000

Model 3_5

	precision	recall	f1-score	support
BENT	0.9715	0.9880	0.9797	1000
COMP	0.9839	0.9780	0.9809	1000
FRI	0.9828	0.9720	0.9774	1000
FRII	0.9530	0.9530	0.9530	1000
avg / total	0.9728	0.9728	0.9727	4000

Model 3_6

	precision	recall	f1-score	support
BENT	0.9752	0.9840	0.9796	1000
COMP	0.9697	0.9930	0.9812	1000
FRI	0.9818	0.9710	0.9764	1000
FRII	0.9673	0.9460	0.9565	1000
avg / total	0.9735	0.9735	0.9734	4000

Model 3_7

	precision	recall	f1-score	support
BENT	0.9900	0.9870	0.9885	1000
COMP	0.9832	0.9940	0.9886	1000
FRI	0.9909	0.9830	0.9869	1000
FRII	0.9710	0.9710	0.9710	1000
avg / total	0.9838	0.9838	0.9837	4000

Model 3_8

	precision	recall	f1-score	support
BENT	0.9910	0.9960	0.9935	1000
COMP	0.9970	0.9860	0.9915	1000
FRI	0.9880	0.9880	0.9880	1000
FRII	0.9761	0.9820	0.9791	1000
avg / total	0.9880	0.9880	0.9880	4000

Model 3_9

	precision	recall	f1-score	support
BENT	0.9930	0.9950	0.9940	1000
COMP	0.9930	0.9920	0.9925	1000
FRI	0.9911	0.9970	0.9940	1000
FRII	0.9839	0.9770	0.9804	1000
avg / total	0.9902	0.9902	0.9902	4000

Model 3_10

	precision	recall	f1-score	support
BENT	0.9660	0.9950	0.9803	1000
COMP	0.9697	0.9910	0.9802	1000
FRI	0.9857	0.9680	0.9768	1000
FRII	0.9741	0.9410	0.9573	1000
avg / total	0.9739	0.9738	0.9736	4000

Model 3_11

	precision	recall	f1-score	support
BENT	0.9773	0.9900	0.9836	1000
COMP	0.9753	0.9860	0.9806	1000
FRI	0.9828	0.9690	0.9758	1000
FRII	0.9616	0.9520	0.9568	1000
avg / total	0.9742	0.9742	0.9742	4000

Model 3_12

	precision	recall	f1-score	support
BENT	0.9840	0.9870	0.9855	1000
COMP	0.9803	0.9960	0.9881	1000
FRI	0.9990	0.9760	0.9874	1000
FRII	0.9661	0.9700	0.9681	1000
avg / total	0.9824	0.9822	0.9823	4000

Model 3_13

	precision	recall	f1-score	support
BENT	0.9840	0.9880	0.9860	1000
COMP	0.9810	0.9860	0.9835	1000
FRI	0.9880	0.9800	0.9840	1000
FRII	0.9770	0.9760	0.9765	1000
avg / total	0.9825	0.9825	0.9825	4000

Model 3_14

	precision	recall	f1-score	support
BENT	0.9980	0.9910	0.9945	1000
COMP	0.9833	1.0000	0.9916	1000
FRI	0.9990	0.9930	0.9960	1000
FRII	0.9869	0.9830	0.9850	1000
avg / total	0.9918	0.9918	0.9918	4000

Model 3_15

	precision	recall	f1-score	support
BENT	0.9960	1.0000	0.9980	1000
COMP	0.9960	0.9990	0.9975	1000
FRI	0.9970	0.9990	0.9980	1000
FRII	0.9980	0.9890	0.9935	1000
avg / total	0.9968	0.9968	0.9967	4000

Model 3_16

	precision	recall	f1-score	support
BENT	0.9880	0.9900	0.9890	1000
COMP	0.9860	0.9880	0.9870	1000
FRI	0.9890	0.9890	0.9890	1000
FRII	0.9870	0.9830	0.9850	1000
avg / total	0.9875	0.9875	0.9875	4000

Model 3_17

	precision	recall	f1-score	support
BENT	0.9880	0.9850	0.9865	1000
COMP	0.9870	0.9900	0.9885	1000
FRI	0.9900	0.9870	0.9885	1000
FRII	0.9820	0.9850	0.9835	1000
avg / total	0.9868	0.9868	0.9868	4000

Model 3_18

	precision	recall	f1-score	support
BENT	0.9970	0.9900	0.9985	1000
COMP	0.9960	0.9980	0.9970	1000
FRI	0.9900	0.9980	0.9990	1000
FRII	0.9960	0.9930	0.9945	1000
avg / total	0.9973	0.9972	0.9972	4000

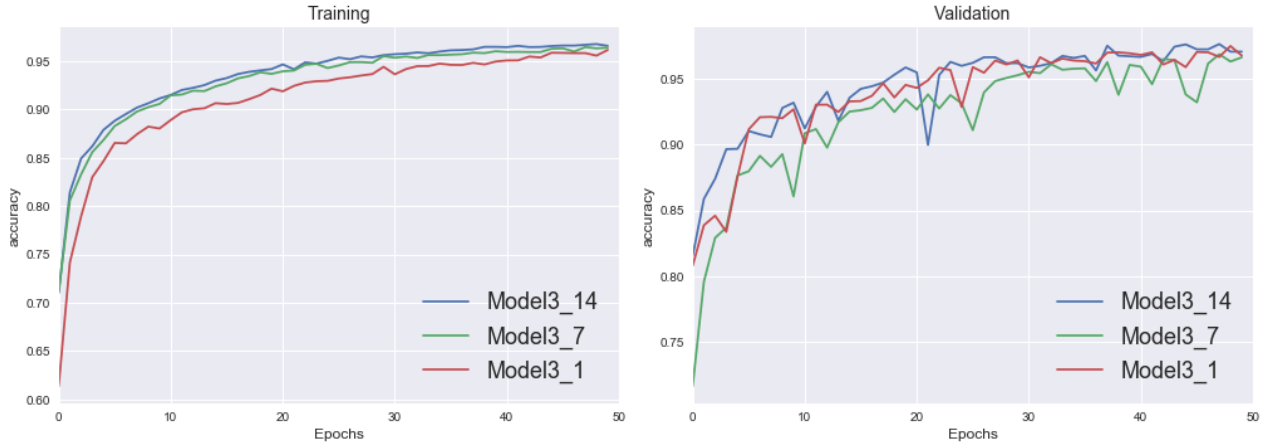
Model 3_19

	precision	recall	f1-score	support
BENT	0.9862	0.9970	0.9915	1000
COMP	0.9939	0.9850	0.9895	1000
FRI	0.9900	0.9880	0.9890	1000
FRII	0.9760	0.9760	0.9760	1000
avg / total	0.9865	0.9865	0.9865	4000

Model 3_20

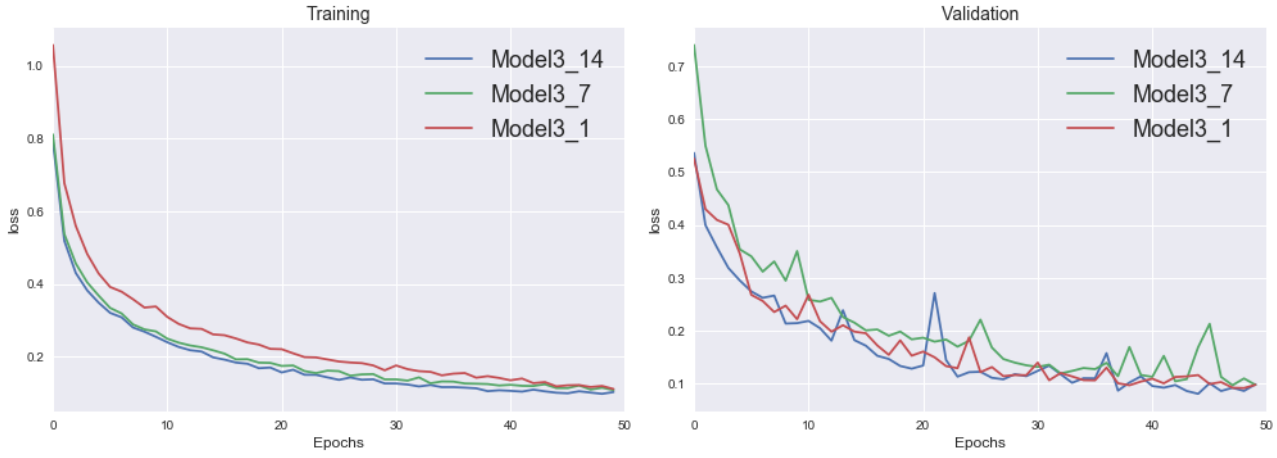
	precision	recall	f1-score	support
BENT	0.9881	0.9990	0.9935	1000
COMP	0.9851	0.9950	0.9900	1000
FRI	0.9910	0.9960	0.9935	1000
FRII	0.9897	0.9640	0.9767	1000
avg / total	0.9885	0.9885	0.9884	4000

Figure 6.10: Model 3 classification reports.



(a) (b)

Figure 6.11: Learning curves of Model 2 (Model 3_1, 3_7 and 3_14) with respect to the number of epochs in the training and validation phase. (a): Training accuracy curves, (b): Testing accuracy curves.



(a) (b)

Figure 6.12: Learning curves of Model 1 (Model 3_1, 3_7 and 3_14) with respect to the number of epochs in the training and validation phase. (a): Training loss curves, (b): Testing loss curves.

6.1.4 Statistical Test for Significance

To determine weather the results from our 3 models are statistically different from each other, we have performed the Friedman test (Friedman, 1937), which is a nonparametric statistical test that tests the null hypothesis ($H0$) that repeated measurements from the same models have the same distribution, in other words, checks if repeated measurements are statically different. Since F1-score is the best measure for models comparison, we have used the F1-score from the 20 runs of each model to perform the test. We used the Friedman test built-in function in python, which takes the measurements and return the statistic and the P-value.

Our $H0$ was that, all the results from our 3 models are the same and there is no significant difference and the significance level α of 0.05. We obtained a P-value of 0.000 for $\alpha = 0.05$ with $Statistic = 40.000$ so we rejected the null hypothesis. This means our results from the 3 models are significantly different.

6.1.5 Results Comparison and Summary

We have trained each of our three models for 20 independent times, and a classification report was produced for each. Figure 6.15, 6.16 and 6.17 show the frequency distribution of the precision, recall and F1-score respectively of each run of the three models. The Friedman statistical test was performed and shows that the results are significantly different. Table 6.1 shows the average values of the precision, recall, f1-score, the standard deviation (STD) and the formal error (σ) on the average values for each of our three models on the testing dataset. Table 6.2 shows the difference in the average values between model 1 and 2, and model 2 and 3, where the statistical error on the difference ($\Delta\sigma$) was calculated for each. Statistically model 2 out performs model 1 with a $\Delta\sigma$ of 0.0021 with ΔP_{avg}, ΔR_{avg} and $\Delta F1_{avg}$ of $\approx$ 0.04. Model 3 is also clearly outperforms model 2 with a $\Delta\sigma$ of 0.0028 with ΔP_{avg}, ΔR_{avg} and $\Delta F1_{avg}$ of ≈ 0.017.

For the fact that we need to choose only one model to use in the classifier to classify our radio sources, we chose 1 model with the highest metrics values from each of our three groups, for the confusion matrices and Loss and accuracy comparison. Model 1_9, model 2_8 and model 3_14 which have an average accuracy of 0.93, 0.97 and 0.99 respectively. These three chosen models are not statistically different from the other runs in their groups, but because eventually we need only one model. Table 6.3 shows the number of miss-classified sources, where Model 3_14 has the least number of miss-classified sources, which has only 48 sources in total were miss-classified. Model 2_8 has 174 miss-classified sources and Model 1_9 has 280 miss-classified sources.

In terms of Loss and accuracy during the training and validation phase, as shown in Figure 6.13 and 6.14, we can see although Model 1_9 and 2_8 have more smoother curves and lower fluctuation than Model 3_14, specially during the validation phase, but still Model 3_14 has higher values of accuracy and smaller values of loss at the end of the training and validation

phase, which confirms the confusion matrices result that model 3_14 has better performance, further testing in terms of quality will be shown in the next Section for Model 3_14 for more robustness. From here on will be only using Model 3_14, which is the model that we used to build the FIRST classifier in Section 7.

Model	P_{avg}	P_{STD}	$\sigma_{P_{avg}}$	R_{avg}	R_{STD}	$\sigma_{R_{avg}}$	$F1_{avg}$	$F1_{STD}$	$\sigma_{F1_{avg}}$
Model 1	0.9263	0.0062	0.0014	0.9257	0.0059	0.0014	0.9256	0.0057	0.0014
Model 2	0.9665	0.0069	0.0016	0.9664	0.0070	0.0016	0.9661	0.0070	0.0016
Model 3	0.9837	0.0101	0.0023	0.9836	0.0102	0.0023	0.9836	0.0102	0.0023

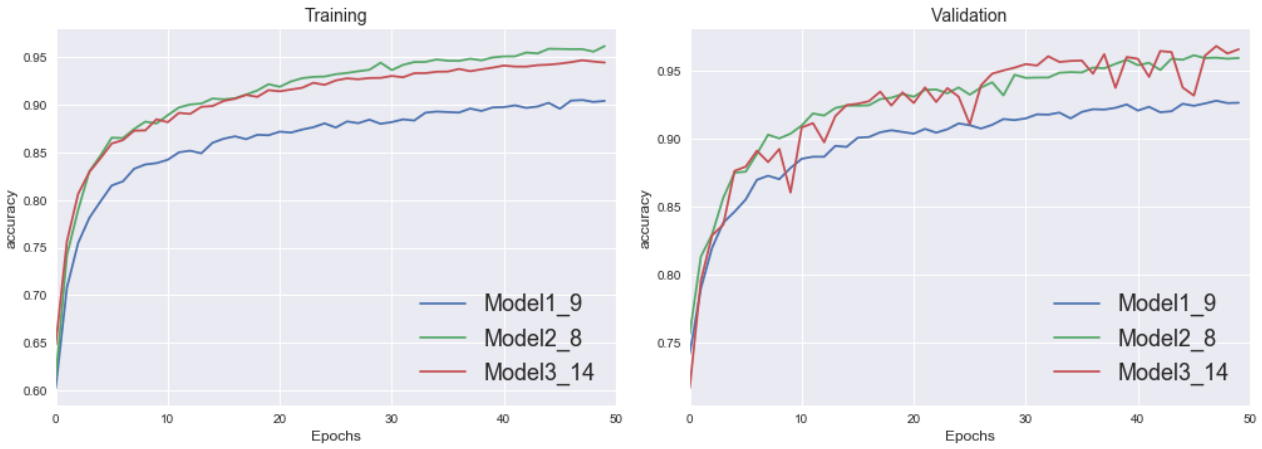
Table 6.1: Comparison of the average (avg), STD and the formal error(σ) of: Precision (P), recall (R) and f1-score (F1) of our 3 models.

Model	ΔP_{avg}	$\Delta\sigma$	ΔR_{avg}	$\Delta\sigma$	$\Delta F1_{avg}$	$\Delta\sigma$
Model 1 - Model 2	0.0402	0.0021	0.0407	0.0021	0.0405	0.0021
Model 2 - Model 3	0.0172	0.0028	0.0172	0.0028	0.0175	0.0028

Table 6.2: Statistical comparison of Model 1,2 and 3.

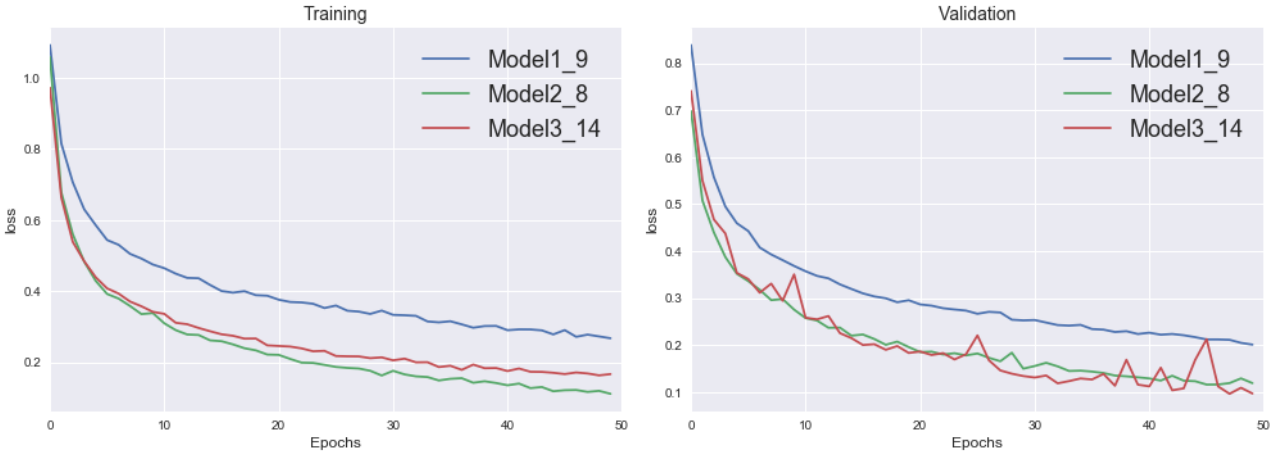
Model	FRIs	FRIIs	Bent-tailed	Compact	Total
Model 1_9	69	129	65	17	280
Model 2_8	9	74	53	38	174
Model 3_14	12	18	4	14	48

Table 6.3: Miss-classified sources of our 3 best performance models in number.



(a) (b)

Figure 6.13: Learning curves of Model 1_9, 2_8 and 3_14 with respect to the number of epochs in the training and validation phase. (a): Training accuracy curves, (b): Testing accuracy curves.



(a) (b)

Figure 6.14: Learning curves of Model 1_9, 2_8 and 3_14 with respect to the number of epochs in the training and validation phase. (a): Training loss curves, (b): Testing loss curves.

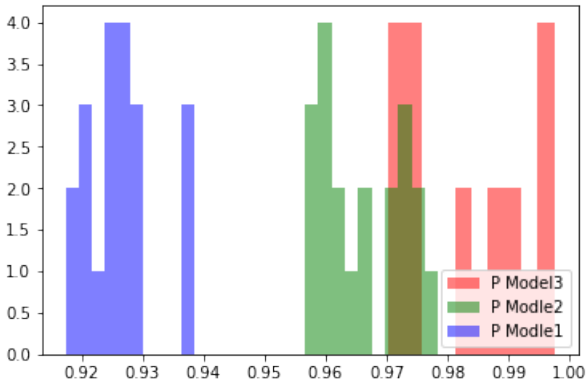


Figure 6.15: Distribution of the *Precision* (P) values from our 3 models.

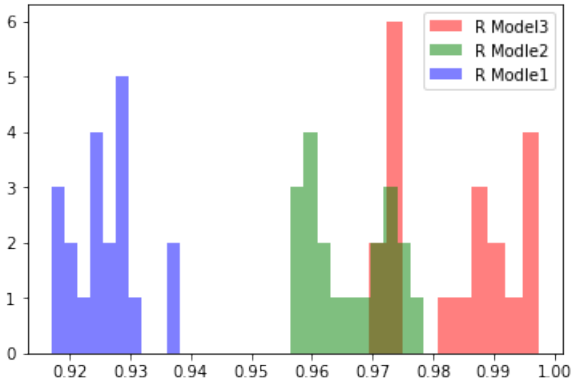


Figure 6.16: Distribution of the *recall* (R) values from our 3 models.

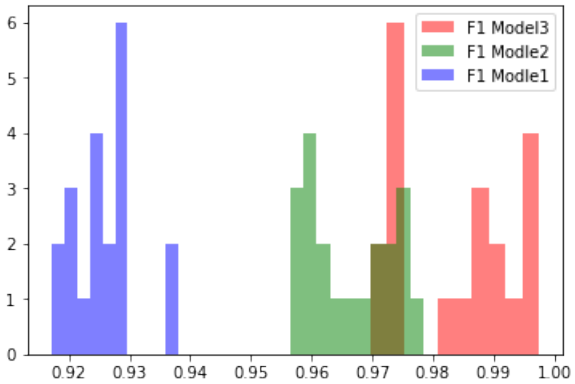


Figure 6.17: Distribution of the $F1 - score$ (F1) values from our 3 models.

6.2 Model 3_14 Qualitative results

To qualitatively evaluate the effectiveness of our best performance CNN model 3_14 in comparison with the ground truth (True labels), 11 random images from our test dataset were classified by our CNN models and visualized as shown in Figure 6.18 and Figure 6.19, along with: predicted class, actual class(ground truth) and the coordinates (Right Ascension and Declination). All the samples for the classification were 100% accurate and no misclassification were spotted, which confirms the quantitative results that model 3 has achieved.

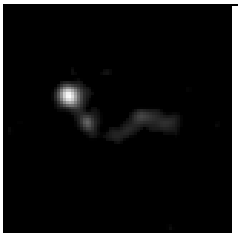
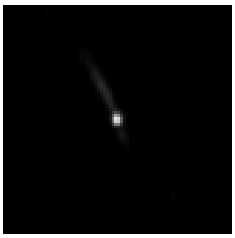
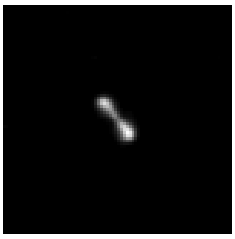
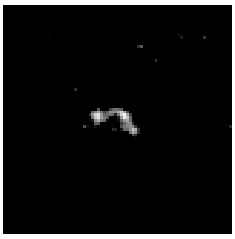
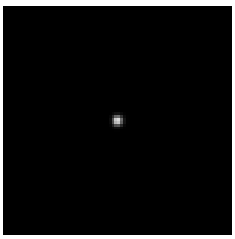
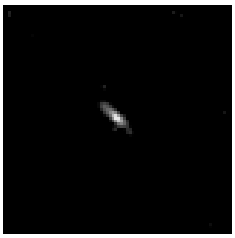
Source Cutout	Ra (deg)	Dec (deg)	True Class	Model Prediction
	162.39066	44.763153	BENT	BENT
	43.658271	0.939403	FR I	FR I
	123.43171	52.96061	FR II	FR II
	138.35497	12.462767	BENT	BENT
	228.97571	24.97789	COMP	COMP
	112.560000	39.533600	FR I	FR I

Figure 6.18: Comparison between our model predictions and ground truth of random samples from FIRST Images

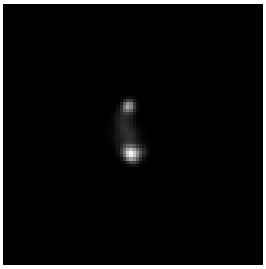
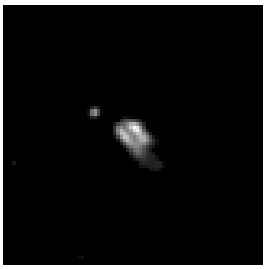
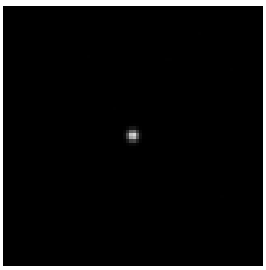
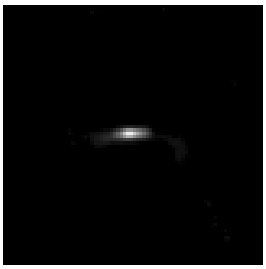
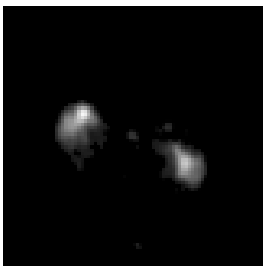
Source Cutout	Ra (deg)	Dec (deg)	True Class	Model Prediction
	123.80137	38.67928	FR II	FR II
	136.56718	44.317738	BENT	BENT
	230.95567	32.23072	COMP	COMP
	113.771567	41.974483	FR I	FR I
	125.69846	4.29722	BENT	BENT

Figure 6.19: Comparison between our model predictions and ground truth of random samples from FIRST Images

Chapter 7

FIRST Classifier

We developed the FIRST classifier based on our CNN model 3_14. The current version of the classifier has access to all the sources available at the final FIRST survey (946,432 sources) (Helfand et al., 2015), which retrieves a postage-stamps from the FIRST archival data using the given coordinate of the source. Each postage-stamp has a well resolved single source which is the input to the model. The classifier consists of two main parts: the pipeline part which features an automatic search for radio sources from FIRST survey archived images (Becker et al., 1995) (which the name FIRST comes) remotely using Virtual Observatory tools (PyVO) (Graham et al., 2014). PyVO provides access to Virtual Observatory services and archived data from different surveys remotely using python. The second part is the CNN model. The FIRST classifier performs the classification through three steps:

- (i) Input: Take a coordinate of the target radio source, Then retrieves cut-out images of the input coordinates from the FIRST archived images.
- (ii) Clean the retrieved image by removing the background noise (remove all the pixels values below 3 σ) and cropping the inner part of the image to a size of 150×150.
- (iii) Feed the source's image to the CNN (model3) to predict its morphology type. This model as mentioned previously is trained to recognize Compact and Extended radio sources. For Extended sources it classifies into three morphology types: FRI, FRII and BENT.
- (iv) Output: predicted morphology type(corresponding to the highest probability), probabilities plot of the classification and a direct link to download the FITS file cut out of the target. If the target source was not found a "data not available error" will be raised, since the image retrieving process is on-line, an error of "Time out please re-run again" will appear in case of weak internet connection.

FIRST Classifier software and instructions may be found at https://github.com/wathela/FIRST-CLASSIFIER. First Classifier has two features, a Single Source classifier and a Multi source classifier.

7.1 Single Source Classification:

Classify only one single source, as shown in Figure 7.1.

- Input: Coordinates of single radio source (Right Ascention and Declination in degree).
- Output: Predicted morphology type(corresponding to the highest probability), probabilities plot of the classification and a direct link to download the FITS file cut out of the target.
- How to run example: `python3 single_source_classification.py` then you will be asked to enter the Ra and Dec of the source.

7.2 Multi Sources Classification:

Allow the classification of a list of sources in a CSV file. Table 7.1 details the output format and columns description of the Multi-source classifier and Figure 7.2 shows an output example produced by it.

- Input: A csv file that has a list of coordinates of sources and index of the Right Ascention and Declination columns (RA and DEC must be in degree).
- Output: A csv file containing 4 columns: Coordinates (RA and DEC), Predicted class, Highest probability, Link to download the cut-out FITS file.
- How to run example:

```
python3 multi_sources_classification.py --data_dir path/to/file/test.csv
 --ra_col 0 --dec_col 1
```

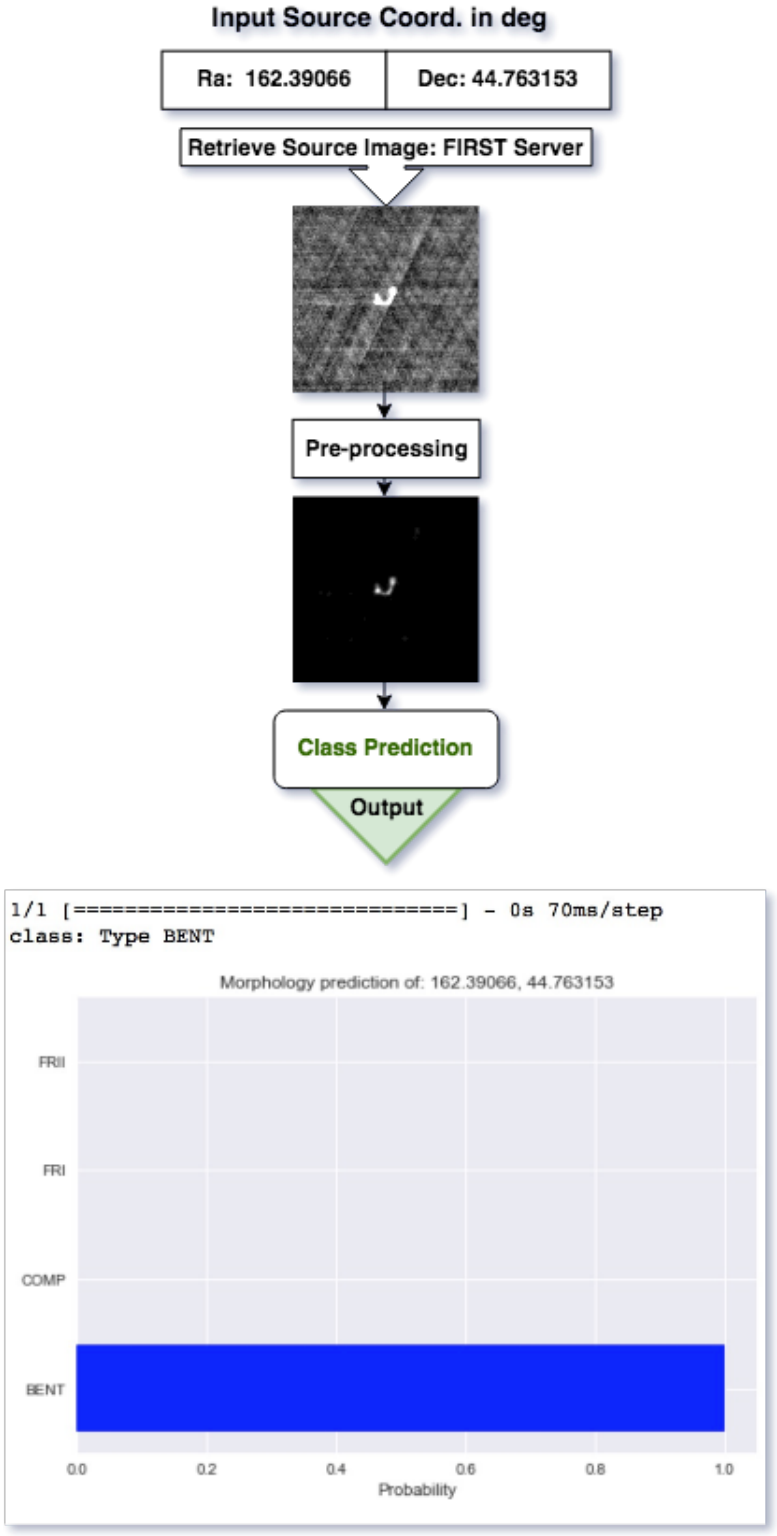


Figure 7.1: Schematic overview of the Single Source classification feature of of the FIRST Classifier.

Column position	Column name	Description
1	FLDNAME	The name of the field from the FIRST catalogue.
2	RA (degrees)	Right ascension (J2000)
3	Dec. (degrees)	Declination (J2000)
4	CLASS	The predicted class, C: Compact, I: FRI, II: FRII, BENT: Bent-tailed.
5	PROB	The probability corresponding to the predicted class, a number between 0 and 1, The nearer to 1 the more relevant the galaxy is to the classification.
6	URL	A direct Link to download the cut-out FITS file.

Table 7.1: The format and columns of the catalogue produced by the Multi-sources classification feature.

	FLDNAME	RA	DEC	CLASS	PROB	URL
2	07360+64243J	239.82	11.2627	II	0.999995	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=239.82003999999998%2C11.26269&survey=first&pixels=300%2C300&sample
3	07480+64243J	212.117	2.43019	I	0.979321	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=212.11725%2C2.43019&survey=first&pixels=300%2C300&sampler=LI&size=0
4	07360+64243J	212.484	-2.76808	I	0.548256	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=212.48447000000002%2C-2.76808&survey=first&pixels=300%2C300&sampler
5	07420+64243J	212.488	-1.35131	I	0.996879	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=212.4875%2C-1.35131&survey=first&pixels=300%2C300&sampler=LI&size=0.
6	07540+64243J	212.768	-3.01203	C	0.732331	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=212.76812999999999%2C-3.0120299999999998&survey=first&pixels=300%2C
7	07480+64243J	212.811	2.28958	I	1	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=212.81087000000002%2C2.28958&survey=first&pixels=300%2C300&sampler=
8	07180+64243J	213.022	-3.32475	C	0.955422	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=213.02196%2C-3.32475&survey=first&pixels=300%2C300&sampler=LI&size=(
9	07000+64243J	213.31	-3.76497	I	0.989326	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=213.30959%2C-3.76497&survey=first&pixels=300%2C300&sampler=LI&size=(
10	07420+64243J	213.312	-3.2075	C	1	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=213.31183%2C-3.2075&survey=first&pixels=300%2C300&sampler=LI&size=0.
11	06540+64243J	213.739	0.20497	BENT	1	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=213.73892%2C0.20496999999999999&survey=first&pixels=300%2C300&samp
12	07420+64243J	226.904	11.0519	II	0.99965	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=226.90417000000002%2C11.05194&survey=first&pixels=300%2C300&sample
13	07420+64243J	227.318	16.9826	II	0.999992	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=227.31812000000002%2C16.98256&survey=first&pixels=300%2C300&sample
14	07180+64243J	227.428	29.6508	II	0.980965	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=227.42833%2C29.650779999999997&survey=first&pixels=300%2C300&sampl
15	07000+64243J	227.68	12.6812	II	1	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=227.6798%2C12.681189999999999&survey=first&pixels=300%2C300&sample
16	07180+64243J	227.705	14.6239	II	0.999865	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=227.70467000000002%2C14.62394&survey=first&pixels=300%2C300&sample
17	07180+64243J	227.873	10.0288	II	0.999997	https://skyview.gsfc.nasa.gov/cgi-bin/images?position=227.87262%2C10.028830000000001&survey=first&pixels=300%2C300&sampl

Figure 7.2: An example of the Catalogue output from the Multi-sources classification feature of the FIRST classifier.

7.3 FIRST Classifier Testing

FR0CAT (Baldi et al., 2018) is a first catalogue includes only FR 0 (compact) radio galaxies,the catalogue combines data from the NVSS, FIRST, and SDSS surveys, and consists of 108 compact radio sources, with redshift $\lesssim 0.05$, radio size $\lesssim 5kpc$, radio luminosities at 1.4GHz in the range $10^{38} \lesssim vL_{1.4} \lesssim 10^{40}\ ergs^{-1}$, and with an optical spectrum characteristic of low-excitation galaxies. 86% of the sources are red early-type galaxies with black hole masses $10^8 \lesssim M_{BH} \lesssim 10^9 M_{\odot}$. Sources in *FR0CAT* represents the dominant population of radio sources in the local Universe.

The reason we did not use the *FR0CAT* to construct and support our compact samples is

that, the catalogue was only released after the work has started already. However, since the samples in the catalogue is not seen by our CNN model and we know to which class all the sources in the catalogue belongs to, it is ideal way to test the performance of the FIRST classifier (the pipeline and the model). We fed the coordinates of all the sources from the $FR0CAT$ catalogue into the classifier. Table 7.2 shows the output classified catalogue, where we can see clearly the accuracy of the classification is 98%. Only 2 sources were classified as not compact sources, the reason seems to be as shown in Figure 7.3, after performing the 150×150 cropping of the inner part of the images, parts of the near by sources were included in the cut-out, which resulted in the miss-classification of the two sources. This results proved the good performance of the pipeline of the classifier in retrieving and cleaning the images, but only show the performance of the model on compact sources, since the testing sources are all compact, which confirms the results our CNN model 3 has achieved for compact sources in chapter 6.

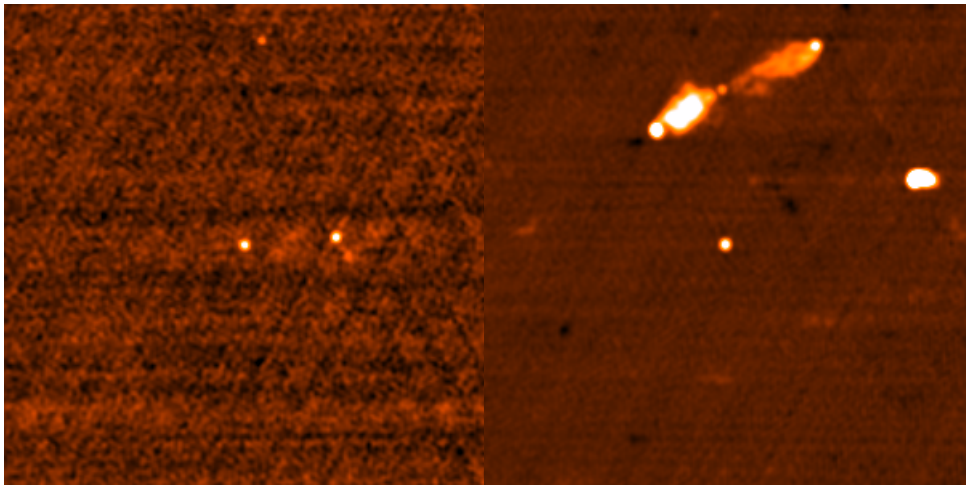
Table 7.2: Re-classification of the FR0 Cataloge using FIRST Classifier

RA	DEC	CLASS	PROB	BENT	COMP	FRI	FRII
17.2187	-0.655206	C	1	0	1	0	0
18.0193	-0.245072	C	1	0	1	0	0
18.8157	0.213419	C	1	0	1	0	0
27.8629	-8.50537	C	1	0	1	0	0
32.1492	-8.63198	C	1	0	1	0	0
118.479	13.1548	C	1	0	1	0	0
121.819	14.9509	C	1	0	1	0	0
128.8	5.3081	C	1	0	1	0	0
130.261	59.9363	C	1	0	1	0	0
131.758	10.0185	C	1	0	1	0	0
136.72	41.4083	C	1	0	1	0	0
136.895	32.9563	C	1	0	1	0	0
137.406	19.4689	C	1	0	1	0	0
137.666	18.6966	C	1	0	1	0	0
139.007	17.5899	C	1	0	1	0	0
139.476	13.5293	C	1	0	1	0	0
142.515	34.2238	C	1	0	1	0	0
143.442	10.1526	C	1	0	1	0	0
144.911	38.8997	I	1	0	0	1	0
145.83	36.2478	C	1	0	1	0	0
151.458	0.633372	C	1	0	1	0	0
153.373	7.90448	C	1	0	1	0	0
154.528	0.0999861	C	1	0	1	0	0

RA	DEC	CLASS	PROB	BENT	COMP	FRI	FRII
156.014	42.1083	C	1	0	1	0	0
156.298	17.2556	C	1	0	1	0	0
156.434	10.375	C	1	0	1	0	0
159.33	43.5875	C	1	0	1	0	0
159.969	20.847	C	1	0	1	0	0
160.118	9.18245	C	1	0	1	0	0
161.015	43.9034	C	1	0	1	0	0
162.05	4.99855	C	1	0	1	0	0
162.22	48.0541	C	1	0	1	0	0
164.38	40.9461	C	1	0	1	0	0
167.805	28.6964	C	1	0	1	0	0
169.095	29.2524	C	1	0	1	0	0
169.251	32.5975	C	1	0	1	0	0
170.122	4.12826	C	1	0	1	0	0
170.166	50.8274	C	1	0	1	0	0
170.735	34.1115	C	1	0	1	0	0
171.605	52.0844	C	1	0	1	0	0
171.865	40.0692	C	1	0	1	0	0
173.694	48.9561	C	1	0	1	0	0
173.706	49.0774	C	1	0	1	0	0
174.155	51.0023	C	1	0	1	0	0
175.629	-2.25151	C	1	0	1	0	0
175.637	26.4889	C	1	0	1	0	0
177.019	37.4439	C	1	0	1	0	0
178.881	54.8668	I	0.377	0	0.36	0.377	0.263
181.465	20.5219	C	1	0	1	0	0
181.532	40.1508	C	1	0	1	0	0
183.372	50.7414	C	1	0	1	0	0
184.965	28.4226	C	1	0	1	0	0
186.089	60.1113	C	1	0	1	0	0
187.55	47.0064	C	1	0	1	0	0
190.828	3.55034	C	1	0	1	0	0
191.641	11.8966	C	1	0	1	0	0
192.614	0.229167	C	1	0	1	0	0
193.538	-1.25757	C	1	0	1	0	0
196.021	7.90796	C	1	0	1	0	0

RA	DEC	CLASS	PROB	BENT	COMP	FRI	FRII
197.158	43.7375	C	1	0	1	0	0
202.677	32.547	C	1	0	1	0	0
203.733	13.742	C	1	0	1	0	0
204.088	3.33099	C	1	0	1	0	0
204.406	15.9722	C	1	0	1	0	0
205.499	29.7815	C	1	0	1	0	0
207.65	33.7049	C	1	0	1	0	0
208.111	14.0911	C	1	0	1	0	0
211.368	30.7672	C	1	0	1	0	0
213.714	3.13083	C	1	0	1	0	0
213.825	-2.44469	C	1	0	1	0	0
216.851	37.4714	C	1	0	1	0	0
217.986	16.7709	C	1	0	1	0	0
218.304	52.9631	C	1	0	1	0	0
218.603	2.7989	C	1	0	1	0	0
219.085	5.33105	C	1	0	1	0	0
221.939	13.3422	C	1	0	1	0	0
223.069	12.2866	C	1	0	1	0	0
223.18	16.9038	C	1	0	1	0	0
224.068	20.5225	C	1	0	1	0	0
225.468	17.7079	C	1	0	1	0	0
226.107	7.82493	C	1	0	1	0	0
226.508	8.78989	C	1	0	1	0	0
226.652	9.43885	C	1	0	1	0	0
227.034	26.9161	C	1	0	1	0	0
230.045	25.7221	C	1	0	1	0	0
230.466	7.70888	C	1	0	1	0	0
232.567	27.0974	C	1	0	1	0	0
235.447	45.556	C	1	0	1	0	0
236.112	47.0067	C	1	0	1	0	0
236.213	43.5142	C	1	0	1	0	0
239.965	25.9407	C	1	0	1	0	0
239.975	44.7089	C	1	0	1	0	0
241.11	17.742	C	1	0	1	0	0
241.349	14.6477	C	1	0	1	0	0
241.567	18.2499	C	1	0	1	0	0

RA	DEC	CLASS	PROB	BENT	COMP	FRI	FRII
241.674	8.74357	C	1	0	1	0	0
243.162	29.6435	C	1	0	1	0	0
243.237	9.86712	C	1	0	1	0	0
245.442	25.8205	C	0.786	0	0.786	0.214	0
246.458	40.4887	C	1	0	1	0	0
247.192	25.4947	C	1	0	1	0	0
247.437	40.8117	C	1	0	1	0	0
252.358	36.0558	C	1	0	1	0	0
254.625	25.3903	C	1	0	1	0	0
255.994	24.1775	C	1	0	1	0	0
258.846	57.4111	C	1	0	1	0	0
260.564	30.7111	C	1	0	1	0	0



(a) (b)

Figure 7.3: Sources classified as FRI, (a): *Ra*:144.911, *Dec*: 38.8997 , (b): *Ra* : 178.881, *Dec*: 54.8668.

Chapter 8

Conclusion and Future Work

8.1 Conclusions

Upcoming surveys with radio observatories like the VLA, ASKAP, MeerKAT and eventually the SKA will generate very large images containing vast numbers of radio galaxies. Different classes of radio galaxies can be used as tracers of the cosmic environment, including dark matter density field, to address key cosmological questions. Manual inspection to systematically classify all the object in these images will be impractical, which motivates developing tools that can automatically analyse them. This includes developing an automatic morphological classification of radio sources.

A CNN model with only three convolutional layers, with batch size of 128, 50 Epochs and Learning Rate of 0.0001 was developed for classifying Compact and Extended radio galaxies observed as part of the the FIRST survey. Our model was trained independently for 20 times and achieved an average accuracy, precision, recall and F1 of 0.98. Based on this model an automatic classifier for radio sources imaged by the FIRST radio survey was developed and presented. The FIRST Classifier is an on-line system for automated classification of FIRST sources, which works well for Compact and Extended Radio Galaxies available on the FIRST image archive. In very rare cases, the double sources in FIRST images might be extended over larger size of the cut-out ($> 150 \times 150$), which implies that, after performing the preprocessing, some of the source might be lost which will results in inaccurate classification. This issue will be solved in the newer coming version of the classifier by developing a source finder that find and extract the target source only from the cut-out given its coordinates, then resize the output image if less or more than 150X150 before it was fed to the CNN model for the prediction.

The FIRST Classifier is publicly available at https://github.com/wathela/FIRST-CLASSIFIER, we have tested it for correctness and robustness, and only basic computer skills are required to use the classifier as a command line utility. Researchers in radio astronomy are encouraged to make use of the classifier and provide us with their feedback about its performance.

This work has been published by Oxford University Press in Monthly Notices for Royal Astronomical Society journal (MNRAS) and available at:
https://academic.oup.com/mnras/advance-article/doi/10.1093/mnras/sty2038/5060783

8.2 Future Work

Future work will focus on generalization of the FIRST classifier to classify sources from different datasets e.g. MeerKAT (MIGHTEE (Taylor & Jarvis, 2017) and MeerKLASS projects), ASKAP (EMU project), GMRT and eventually SKA. This can be done by acquiring labelled samples from these datasets and train the model on, which need to be done manually.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, software available from tensorflow.org

Agarap, A. F. 2017, CoRR, abs/1712.03541

Aird, J., Coil, A. L., Moustakas, J., Blanton, M. R., Burles, S. M., Cool, R. J., Eisenstein, D. J., Smith, M. S. M., Wong, K. C., & Zhu, G. 2011

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., Bengio, Y., Bergeron, A., Bergstra, J., Bisson, V., Bleecher Snyder, J., Bouchard, N., Boulanger-Lewandowski, N., Bouthillier, X., de Brébisson, A., Breuleux, O., Carrier, P.-L., Cho, K., Chorowski, J., Christiano, P., Cooijmans, T., Côté, M.-A., Côté, M., Courville, A., Dauphin, Y. N., Delalleau, O., Demouth, J., Desjardins, G., Dieleman, S., Dinh, L., Ducoffe, M., Dumoulin, V., Ebrahimi Kahou, S., Erhan, D., Fan, Z., Firat, O., Germain, M., Glorot, X., Goodfellow, I., Graham, M., Gulcehre, C., Hamel, P., Harlouchet, I., Heng, J.-P., Hidasi, B., Honari, S., Jain, A., Jean, S., Jia, K., Korobov, M., Kulkarni, V., Lamb, A., Lamblin, P., Larsen, E., Laurent, C., Lee, S., Lefrancois, S., Lemieux, S., Léonard, N., Lin, Z., Livezey, J. A., Lorenz, C., Lowin, J., Ma, Q., Manzagol, P.-A., Mastropietro, O., McGibbon, R. T., Memisevic, R., van Merriënboer, B., Michalski, V., Mirza, M., Orlandi, A., Pal, C., Pascanu, R., Pezeshki, M., Raffel, C., Renshaw, D., Rocklin, M., Romero, A., Roth, M., Sadowski, P., Salvatier, J., Savard, F., Schlüter, J., Schulman, J., Schwartz, G., Serban, I. V., Serdyuk, D., Shabanian, S., Simon, E., Spieckermann, S., Subramanyam, S. R., Sygnowski, J., Tanguay, J., van Tulder, G., Turian, J., Urban, S., Vincent, P., Visin, F., de Vries, H., Warde-Farley, D., Webb, D. J., Willson, M., Xu, K., Xue, L., Yao, L., Zhang, S., & Zhang, Y. 2016, arXiv e-prints, abs/1605.02688

Aniyan, A. K. & Thorat, K. 2017, The Astrophysical Journal Supplement Series, 230, 20

Bai, J., Chen, Z., Feng, B., & Xu, B. 2014, in 2014 IEEE International Conference on Image Processing (ICIP) (IEEE), 2560–2564

Baldi, R. D. & Capetti, A. 2010, Astronomy and Astrophysics, 519, A48

Baldi, R. D., Capetti, A., & Giovannini, G. 2016, Astronomische Nachrichten, 337, 114

Baldi, R. D., Capetti, A., & Massaro, F. 2018, Astronomy & Astrophysics, 609, A1

Ballard, D. H. D. H. & Brown, C. M. 1982, Computer vision (Prentice-Hall), 523

Banfield, J. K., Wong, O. I., Willett, K. W., Norris, R. P., Rudnick, L., Shabala, S. S., Simmons, B. D., Snyder, C., Garon, A., Seymour, N., Middelberg, E., Andernach, H., Lintott, C. J., Jacob, K., Kapińska, A. D., Mao, M. Y., Masters, K. L., Jarvis, M. J., Schawinski, K., Paget, E., Simpson, R., Klöckner, H.-R., Bamford, S., Burchell, T., Chow, K. E., Cotter, G., Fortson, L., Heywood, I., Jones, T. W., Kaviraj, S., López-Sánchez, Á. R., Maksym, W. P., Polsterer, K., Borden, K., Hollow, R. P., & Whyte, L. 2015, Monthly Notices of the Royal Astronomical Society, 453, 2327

Becker, R. H., White, R. L., & Helfand, D. J. 1995, The Astrophysical Journal, 450, 559

Bengio, Y. 2009, Foundations and Trends® in Machine Learning, 2, 1

Bennett, A. S. & Simth, F. G. 1962, Monthly Notices of the Royal Astronomical Society, 125, 75

Best, P. N. & Heckman, T. M. 2012, MNRAS, 421, 1569

Blanton, E. L. & Gregg, M. D. 2003, The Astronomical Journal, 125, 1635

Blanton, E. L., Gregg, M. D., Helfand, D. J., Becker, R. H., & Leighly, K. M. 2001, The Astronomical Journal, 121, 2915

Blanton, E. L., Gregg, M. D., Helfand, D. J., Becker, R. H., & White, R. L. 2000, The Astrophysical Journal, 531, 118

Bonfield, D. G., Sun, Y., Davey, N., Jarvis, M. J., Abdalla, F. B., Banerji, M., & Adams, R. G. 2009

Bose, N. K. N. K. & Liang, P. 1996, Neural network fundamentals with graphs, algorithms, and applications (McGraw-Hill), 478

Braun, R., Bourke, T., Green, J. A., Keane, E., & Wagg, J. 2015, Advancing Astrophysics with the Square Kilometre Array (AASKA14), 174

Bridle, J. S. 1990, in Neurocomputing, ed. F. F. Soulié & J. Hérault (Berlin, Heidelberg: Springer Berlin Heidelberg), 227–236

Buta, Ronald J. 2011, eprint arXiv:1102.0550, 6, 1

Capetti, A. & et al. 2017, Astronomy & Astrophysics, 601, A81

Capetti, A., Massaro, F., & Baldi, R. D. 2016, Astronomy & Astrophysics, 598, A49

Chollet, F. et al. 2015, Keras, https://github.com/keras-team/keras

Collister, A. A. & Lahav, O. 2004, Publications of the Astronomical Society of the Pacific, 116, 345

Collobert, R., Bengio, S., & Marithoz, J. 2002, Torch: A Modular Machine Learning Software Library

Condon, J. J., Cotton, W. D., Greisen, E. W., Yin, Q. F., Perley, R. A., Taylor, G. B., & Broderick, J. J. 1998, The Astronomical Journal, 8065, 1693

Condon, J. J. & J., J. 1992, Annual Review of Astronomy and Astrophysics, 30, 575

Cybenko, G. 1989, Mathematics of Control, Signals, and Systems, 2, 303

Davim, J. P. 2011, Machining of hard materials (Springer), 211

de la Calleja, J. & Fuentes, O. 2004, MNRAS, 349, 87

Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441

D'Isanto, A. & Polsterer, K. L. 2018, Astronomy & Astrophysics, 609, A111

Douglass, E. M., Blanton, E. L., Clarke, T. E., Randall, S. W., & Wing, J. D. 2011, The Astrophysical Journal, 743, 199

Douglass, E. M., Blanton, E. L., Clarke, T. E., Sarazin, C. L., & Wise, M. 2007, The Astrophysical Journal, 673, 763

Dreyfus, S. 1962, Journal of Mathematical Analysis and Applications, 5, 30

Escalera, S., Baró, X., Pujol, O., Vitrià, J., & Radeva, P. 2011, 15–52

Fanaroff, B. L. & Riley, J. M. 1974, MNRAS, 167, 31P

Fawcett, T. 2006, Pattern Recognition Letters, 27, 861 , rOC Analysis in Pattern Recognition

Friedman, M. 1937, Journal of the American Statistical Association, 32, 675

Fukushima, K. 1980, Biological Cybernetics, 36, 193

Gendre, M. A., Best, P. N., & Wall, J. V. 2010, MNRAS, 404, 1719

Gendre, M. A., Best, P. N., Wall, J. V., & Ker, L. M. 2013, Monthly Notices of the Royal Astronomical Society, 430, 3086

Gendre, M. A. & Wall, J. V. 2009, MNRAS, 394, 1712

Gers, F. & Schmidhuber, E. 2001, IEEE Transactions on Neural Networks, 12, 1333

Glorot, X., Bordes, A., & Bengio, Y. 2011, AISTATS '11: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, 15, 315

Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (MIT Press), http://www.deeplearningbook.org

Graham, M., Plante, R., Tody, D., & Fitzpatrick, M. 2014, PyVO: Python access to the Virtual Observatory, Astrophysics Source Code Library

Guo, Y., Giavalisco, M., Cassata, P., Ferguson, H. C., Williams, C. C., Dickinson, M., Koekemoer, A. M., Grogin, N. A., Chary, R.-R., Messias, H., Tundo, E., Lin, L., Lee, S.-K., Salimbeni, S., Fontana, A., Grazian, A., Kocevski, D., Lee, K.-S., Villanueva, E., & van der Wel, A. 2011

Hahnioser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. 2000, Nature, 405, 947

Hecht-Nielsen, R. 1988, Neural Networks, 1, 445

Helfand, D. J., White, R. L., & Becker, R. H. 2015, The Astrophysical Journal, 801, 26

Hey, J. S. 1983, The radio universe (Pergamon Press), 246

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2012, CoRR, abs/1207.0580

Hocking, A., Geach, J. E., Davey, N., & Sun, Y. 2015, ArXiv e-prints

Hopkins, A. M., Whiting, M. T., Seymour, N., Chow, K. E., Norris, R. P., Bonavera, L., Breton, R., Carbone, D., Ferrari, C., Franzen, T. M. O., Garsden, H., González-Nuevo, J., Hales, C. A., Hancock, P. J., Heald, G., Herranz, D., Huynh, M., Jurek, R. J., López-Caniego, M., Massardi, M., Mohan, N., Molinari, S., Orrù, E., Paladino, R., Pestalozzi, M., Pizzo, R., Rafferty, D., Röttgering, H. J. A., Rudnick, L., Schisano, E., Shulevski, A., Swinbank, J., Taylor, R., & van der Horst, A. J. 2015, Publications of the Astronomical Society of Australia, 32, e037

Hubble, E. P. & P., E. 1926, The Astrophysical Journal, 64, 321

Ivezic, Z., Connolly, A. J., VanderPlas, J. T., & Gray, A. 2014, Statistics, data mining, and machine learning in astronomy: a practical Python guide for the analysis of survey data, Princeton Series in Modern Observational Astronomy (Princeton, NJ: Princeton University Press)

Jansky, K. 1933, Proceedings of the IRE, 21, 1387

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. 2014, arXiv preprint arXiv:1408.5093

Jiang, H. & Learned-Miller, E. 2016, in 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017) (IEEE), 650–657

Johnston, S., Bailes, M., Bartel, N., Baugh, C., Bietenholz, M., Blake, C., Braun, R., Brown, J., Chatterjee, S., Darling, J., Deller, A., Dodson, R., Edwards, P. G., Ekers, R., Ellingsen, S., Feain, I., Gaensler, B. M., Haverkorn, M., Hobbs, G., Hopkins, A., Jackson, C., James, C., Joncas, G., Kaspi, V., Kilborn, V., Koribalski, B., Kothes, R., Landecker, T. L., Lenc, E., Lovell, J., Macquart, J.-P., Manchester, R., Matthews, D., McClure-Griffiths, N. M., Norris, R., Pen, U.-L., Phillips, C., Power, C., Protheroe, R., Sadler, E., Schmidt, B., Stairs, I., Staveley-Smith, L., Stil, J., Taylor, R., Tingay, S., Tzioumis, A.,

Walker, M., Wall, J., & Wolleben, M. 2007, Publications of the Astronomical Society of Australia, 24, 174

Johnston, S., Taylor, R., Bailes, M., Bartel, N., Baugh, C., Bietenholz, M., Blake, C., Braun, R., Brown, J., Chatterjee, S., Darling, J., Deller, A., Dodson, R., Edwards, P., Ekers, R., Ellingsen, S., Feain, I., Gaensler, B., Haverkorn, M., Hobbs, G., Hopkins, A., Jackson, C., James, C., Joncas, G., Kaspi, V., Kilborn, V., Koribalski, B., Kothes, R., Landecker, T., Lenc, A., Lovell, J., Macquart, J.-P., Manchester, R., Matthews, D., McClure-Griffiths, N., Norris, R., Pen, U.-L., Phillips, C., Power, C., Protheroe, R., Sadler, E., Schmidt, B., Stairs, I., Staveley-Smith, L., Stil, J., Tingay, S., Tzioumis, A., Walker, M., Wall, J., & Wolleben, M. 2008, Experimental Astronomy, 22, 151

Jonas, J. & MeerKAT Team. 2016, in Proceedings of MeerKAT Science: On the Pathway to the SKA. 25-27 May, 2016 Stellenbosch, South Africa (MeerKAT2016). Online at ¡A href="href="¿href="https://pos.sissa.it/cgi-bin/reader/conf.cgi?confid=277¡/A¿, id.1, 1

Kawaguchi, K. 2000, ETD Collection for University of Texas, El Paso

Kelley, H. J. 1960, ARS Journal, 30, 947

Kim, E. J. & Brunner, R. J. 2017, Monthly Notices of the Royal Astronomical Society, 464, 4463

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, Advances In Neural Information Processing Systems, 1

Lahav, O., Naim, A., Buta, R. J., Corwin, H. G., de Vaucouleurs, G., Dressler, A., Huchra, J. P., van den Bergh, S., Raychaudhury, S., Sodre, Jr., L., & Storrie-Lombardi, M. C. 1995, Science, 267, 859

LeCun, Y., Bengio, Y., & Hinton, G. 2015, Nature, 521, 436

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998, Proceedings of the IEEE, 86, 2278

LeCun, Y., Kavukcuoglu, K., & Farabet, C. 2010, in Proceedings of 2010 IEEE International Symposium on Circuits and Systems (IEEE), 253–256

Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., Murray, P., & Vandenberg, J. 2008, MNRAS, 389, 1179

Lukic, V. & Brüggen, M. 2016, Proceedings of the International Astronomical Union, 12, 217

Lukic, V., Brüggen, M., Banfield, J. K., Wong, O. I., Rudnick, L., Norris, R. P., & Simmons, B. 2018, MNRAS, 476, 246

Ma, W. & Lu, J. 2017, ArXiv e-prints

Machart, P. & Ralaivola, L. 2012, working paper or preprint

Madau, P. & Dickinson, M. 2014, Annual Review of Astronomy and Astrophysics, 52, 415

Makhathini, S., Jarvis, M., Smirnov, O., & Heywood, I. 2015, Advancing Astrophysics with the Square Kilometre Array (AASKA14), 81

Massardi, M., Ekers, R. D., Murphy, T., Mahony, E., Hancock, P. J., Chhetri, R., De Zotti, G., Sadler, E. M., Burke-Spolaor, S., Calabretta, M., Edwards, P. G., Ekers, J. A., Jackson, C. A., Kesteven, M. J., Newton-McGee, K., Phillips, C., Ricci, R., Roberts, P., Sault, R. J., Staveley-Smith, L., Subrahmanyan, R., Walker, M. A., & Wilson, W. E. 2010

McCulloch, W. S. & Pitts, W. 1943, The Bulletin of Mathematical Biophysics, 5, 115

Naim, A., Ratnatunga, K. U., & Griffiths, R. E. 1997, The Astrophysical Journal, 476, 510

Nair, V. & Hinton, G. E. 2010, in Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10 (USA: Omnipress), 807–814

Naumov, M. 2017, ArXiv e-prints

Owen, F. N. & Laing, R. A. 1989, MNRAS, 238, 357

Padovani, P. 2011, Monthly Notices of the Royal Astronomical Society, 411, 1547

Park, S. & Kwak, N. 2017 (Springer, Cham), 189–204

Patil, J. S., Pawase, R. S., & Dandawate, Y. H. 2017, in 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (IEEE), 1168–1172

Polsterer, K. L., Gieseke, F., & Igel, C. 2015, in Astronomical Society of the Pacific Conference Series, Vol. 495, Astronomical Data Analysis Software an Systems XXIV (ADASS XXIV), ed. A. R. Taylor & E. Rosolowsky, 81

Polsterer, K. L., Gieseke, F., & Igel, C. 2015, Astronomical Data Analysis Software and Systems: Xxiv, 495, 81

Proctor, D. D. 2011, The Astrophysical Journal Supplement Series, 194, 31

Rosenblatt, F. 1958, Psychological Review, 65

Royal Astronomical Society of Canada., J. D. & Chant, C. A. 1988, The Journal of the Royal Astronomical Society of Canada., Vol. 82 (Royal Astronomical Society of Canada), 107

Ruder, S. 2016, ArXiv e-prints, 14

Rybicki, G. B. & Lightman, A. P., eds. 2004, Radiative Processes in Astrophysics (Weinheim, Germany: Wiley-VCH Verlag GmbH)

Ryle, M., Smith, F. G., & Elsmore, B. 1950, Monthly Notices of the Royal Astronomical Society, 110, 508

Sadler, E. M., Jackson, C. A., Cannon, R. D., McIntyre, V. J., Murphy, T., Bland-Hawthorn, J., Bridges, T., Cole, S., Colless, M., Collins, C., Couch, W., Dalton, G., De Propris, R., Driver, S. P., Efstathiou, G., Ellis, R. S., Frenk, C. S., Glazebrook, K., Lahav, O., Lewis, I., Lumsden, S., Maddox, S., Madgwick, D., Norberg, P., Peacock, J. A., Peterson, B. A., Sutherland, W., & Taylor, K. 2002, Monthly Notices of the Royal Astronomical Society, 329, 227

Sadler, E. M., Jenkins, C. R., & Kotanyi, C. G. 1989, Monthly Notices of the Royal Astronomical Society, 240, 591

Sakelliou, I. & Merrifield, M. R. 2000, Monthly Notices of the Royal Astronomical Society, 311, 649

Sandage, A. & Allan. 1965, The Astrophysical Journal, 141, 1560

Sandage, A., S. M. & Kristian, J. 1975, in Galaxies and the Universe, Stars and Stellar Systems, Vol. IX, 847

Santos, M., Bull, P., Camera, S., Chen, S., Fonseca, J., Heywood, I., Hilton, M., Jarvis, M., Jozsa, G. I. G., Knowles, K., Leeuw, L., Maartens, R., Malefahlo, E., McAlpine, K., Moodley, K., Patel, P., Pourtsidou, A., Prescott, M., Spekkens, K., Taylor, R., Witzemann, A., & Whittam, I. H. 2017, in Proceedings of MeerKAT Science: On the Pathway to the SKA — PoS(MeerKAT2016), Vol. 277 (Trieste, Italy: Sissa Medialab), 032

Scherer, D., Müller, A., & Behnke, S. 2010, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6354 LNCS, 92

Schmidhuber, J. 2015, Neural Networks, 61, 85

Searle, L. & Sargent, W. L. W. 1972, The Astrophysical Journal, 173, 25

Selim, I., E., A., & M.El, B. 2016, International Journal of Computer Applications, 137, 4

Selim, I. M. & Abd El Aziz, M. 2017, Experimental Astronomy, 43, 131

Seyfert, C. K. 1943, The Astrophysical Journal, 97, 28

Shakeshaft, J. R., Ryle, M., Baldwin, J. E., Elsmore, B., & Thomson, J. H. 1955, Memoirs of the Royal Astronomical Society, Vol. 67, p.106, 67, 106

Simonyan, K. & Zisserman, A. 2014, ArXiv e-prints

Sparke, L. S. & Gallagher, J. S. J. S. 2007, Galaxies in the universe : an introduction (Cambridge University Press), 431

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, Journal of Machine Learning Research, 15

Staravoitau, A. 2018, Pattern Recognition and Image Analysis, 28, 155

Stehman, S. V. 1997, Remote Sensing of Environment, 62, 77

Storrie-Lombardi, M. C., Lahav, O., Sodre, Jr., L., & Storrie-Lombardi, L. J. 1992, MNRAS, 259, 8P

Taylor, A. R. & Jarvis, M. 2017, IOP Conference Series: Materials Science and Engineering, 198, 012014

Urry, M. J. & Sollich, P. 2012, Journal of Physics A: Mathematical and Theoretical, 45, 425005

Vijendra, S., Vasudeva, N., & Parashar, H. J. 2011, CoRR, abs/1612.00625

Werbos, P. J. 1982, in System Modeling and Optimization (Berlin/Heidelberg: Springer-Verlag), 762–770

White, R. L., Becker, R. H., Helfand, D. J., & Gregg, M. D. 1997, The Astrophysical Journal, 475, 479

Wojna, Z., Gorban, A., Lee, D.-S., Murphy, K., Yu, Q., Li, Y., & Ibarz, J. 2017, ArXiv e-prints

York, D. G., Adelman, J., Anderson, Jr., J. E., Anderson, S. F., Annis, J., Bahcall, N. A., Bakken, J. A., Barkhouser, R., Bastian, S., Berman, E., Boroski, W. N., Bracker, S., Briegel, C., Briggs, J. W., Brinkmann, J., Brunner, R., Burles, S., Carey, L., Carr, M. A., Castander, F. J., Chen, B., Colestock, P. L., Connolly, A. J., Crocker, J. H., Csabai, I., Czarapata, P. C., Davis, J. E., Doi, M., Dombeck, T., Eisenstein, D., Ellman, N., Elms, B. R., Evans, M. L., Fan, X., Federwitz, G. R., Fiscelli, L., Friedman, S., Frieman, J. A., Fukugita, M., Gillespie, B., Gunn, J. E., Gurbani, V. K., de Haas, E., Haldeman, M., Harris, F. H., Hayes, J., Heckman, T. M., Hennessy, G. S., Hindsley, R. B., Holm, S., Holmgren, D. J., Huang, C.-h., Hull, C., Husby, D., Ichikawa, S.-I., Ichikawa, T., Ivezić, Ž., Kent, S., Kim, R. S. J., Kinney, E., Klaene, M., Kleinman, A. N., Kleinman, S., Knapp, G. R., Korienek, J., Kron, R. G., Kunszt, P. Z., Lamb, D. Q., Lee, B., Leger, R. F., Limmongkol, S., Lindenmeyer, C., Long, D. C., Loomis, C., Loveday, J., Lucinio, R., Lupton, R. H., MacKinnon, B., Mannery, E. J., Mantsch, P. M., Margon, B., McGehee, P., McKay, T. A., Meiksin, A., Merelli, A., Monet, D. G., Munn, J. A., Narayanan, V. K., Nash, T., Neilsen, E., Neswold, R., Newberg, H. J., Nichol, R. C., Nicinski, T., Nonino, M., Okada, N., Okamura, S., Ostriker, J. P., Owen, R., Pauls, A. G., Peoples, J., Peterson, R. L., Petravick, D., Pier, J. R., Pope, A., Pordes, R., Prosapio, A., Rechenmacher, R., Quinn, T. R., Richards, G. T., Richmond, M. W., Rivetta, C. H., Rockosi, C. M., Ruthmansdorfer, K., Sandford, D., Schlegel, D. J., Schneider, D. P., Sekiguchi, M., Sergey, G., Shimasaku, K., Siegmund, W. A., Smee, S., Smith, J. A., Snedden, S., Stone, R., Stoughton, C., Strauss, M. A., Stubbs, C., SubbaRao, M., Szalay, A. S., Szapudi, I., Szokoly, G. P., Thakar, A. R., Tremonti, C., Tucker, D. L., Uomoto, A., Vanden Berk, D., Vogeley, M. S., Waddell, P., Wang, S.-i., Watanabe, M., Weinberg, D. H., Yanny, B., Yasuda, N., & SDSS Collaboration. 2000, AJ, 120, 1579

Zhang, Y., Li, L., & Zhao, Y. 2009, Monthly Notices of the Royal Astronomical Society, 392, 233