



Classifying Radio Galaxies with the Convolutional Neural Network

A. K. Aniyan^{1,2} and K. Thorat^{1,2}

¹ Department of Physics and Electronics, Rhodes University, Grahamstown, South Africa

² SKA South Africa, 3rd Floor, The Park, Cape Town, South Africa

Received 2017 January 24; revised 2017 May 9; accepted 2017 May 11; published 2017 June 13

Abstract

We present the application of a deep machine learning technique to classify radio images of extended sources on a morphological basis using convolutional neural networks (CNN). In this study, we have taken the case of the Fanaroff–Riley (FR) class of radio galaxies as well as radio galaxies with bent-tailed morphology. We have used archival data from the Very Large Array (VLA)—Faint Images of the Radio Sky at Twenty Centimeters survey and existing visually classified samples available in the literature to train a neural network for morphological classification of these categories of radio sources. Our training sample size for each of these categories is ~ 200 sources, which has been augmented by rotated versions of the same. Our study shows that CNNs can classify images of the FRI and FR II and bent-tailed radio galaxies with high accuracy (maximum precision at 95%) using well-defined samples and a “fusion classifier,” which combines the results of binary classifications, while allowing for a mechanism to find sources with unusual morphologies. The individual precision is highest for bent-tailed radio galaxies at 95% and is 91% and 75% for the FRI and FR II classes, respectively, whereas the recall is highest for FRI and FR IIs at 91% each, while the bent-tailed class has a recall of 79%. These results show that our results are comparable to that of manual classification, while being much faster. Finally, we discuss the computational and data-related challenges associated with the morphological classification of radio galaxies with CNNs.

Key words: methods: miscellaneous – methods: observational – radio continuum: galaxies – techniques: miscellaneous

Supporting material: machine-readable table

1. Introduction

With the advent of the Square Kilometer Array (SKA) radio telescope along with its precursor facilities, we expect the radio sky to be surveyed at high speed and to unprecedented sensitivity. While this may enable paradigm shifts in the studies of radio sources, it comes with very high data volumes. For example, the typical image size from the MeerKAT telescope is estimated to be 11.13 TB³ (individual MeerKat surveys expect to deal with a large number of objects, e.g., the MeerKlass survey expects to find upwards of 200,000 radio sources in the H I emission line⁴). This introduces challenges in all the steps of data reduction from RFI mitigation to calibration and imaging. The science level data volume is expected to be similarly formidable. For example, extrapolating from the Square Kilometre Array Design Studies (Wilman et al. 2010), gives a source density of 6.2×10^4 sources per square degree for a survey reaching 1 microJy at 1.4 GHz (Padovani 2016), and the Evolutionary Map of the Universe survey with Australian Square Kilometer Array Pathfinder (Norris et al. 2011) is expected to find about 70 million radio sources, while covering two-thirds of the sky. Handling this amount of data is not possible through manual studies: automation of data processing is therefore essential.

In this study, we consider the case of radio galaxy classification in the image domain. Traditionally, source detection and classification is trivially done for unresolved and slightly resolved radio sources through various source finding software (these radio sources may in fact be

“components” rather than true sources, for example, these might be lobes of a double radio galaxy). Somewhat more recently, there have been several attempts to classify/identify radio sources through automated techniques (crowd-sourcing is an alternative, e.g., Radio Galaxy Zoo Banfield et al. 2015) such as pattern recognition and decision trees (Proctor 2011), source matching and pattern recognition (van Velzen et al. 2015) and self-organizing maps (Polsterer et al. 2015). The latter is an example of a machine learning technique, which has come into increased use in recent years (especially in pulsar and transient detection/identification, see Eatough et al. 2010; Bates et al. 2012; Morello et al. 2014; Wagstaff et al. 2016).

The typical process of source detection and classification hinges on using source finding software to generate source component catalogs. These components need to be combined (or identified) as a single source, where necessary (source deblending would be the opposite issue). This is especially important for extended sources, which are more likely to get divided in multiple components. These can be AGN-powered or star-forming galaxies. In the past, studies have tended to classify these sources by visual examination. This quickly grows impracticable with increasing survey sizes. Here, we consider the application of deep machine learning techniques to classify extended extragalactic radio sources, more specifically, AGN-powered radio galaxies.

Machine learning methods have been applied to a variety of astronomical problems, such as star–galaxy classification (Weir et al. 1995), redshift estimation (Benitez 2000), classification of optical transients (Mahabal et al. 2011), and unsupervised source segmentation (Hocking et al. 2015) among others. These methods have been robust and reliable, and have

³ MeerKat SDP group, S. Ratcliffe, B. Merry, and T. Bennett.

⁴ Józsa, G.I.G., SKA South Africa, private communication.

performed with high accuracy. For example, with classification of stars and galaxies the best available model has over 99% accuracy (Kim & Brunner 2017). Estimation of redshifts with machine learning has been done with accuracies over 92% (Cavuoti et al. 2017a, 2017b). Machine learning methods have been incorporated into real-time transient classification systems and they perform with accuracies of more than 90% (Mahabal et al. 2008, 2011).

Classic machine learning algorithms such as support vector machines, K-nearest neighbors, and decision trees generally learn from “features” extracted from the observational data (Kotsiantis et al. 2007). Features represent unique characteristics of the raw data and are domain specific. Feature extraction is carefully done so that the chosen features will represent specific physical properties of the system (Guyon & Elisseeff 2006). The efficiency of the learning algorithm mainly depends on the quality of the features used (Blum & Langley 1997). Such machine learning algorithms are generally called shallow learning methods (Chen 1995). In principle, shallow learning methods learn from the features rather than the raw data, which may be images or time series values. A good understanding of the data and the objective under investigation is required to properly extract the features and fine tune the machine learning algorithm. The features extracted also may not encapsulate the distinct properties of the object.

Deep learning is a branch of machine learning in which the machine learning algorithms learn directly from the data instead of features (Bengio 2009). Deep learning is advantageous in situations where engineered features do not completely capture the physics of the raw data and the machine learning algorithm is not able to learn with minimal loss (Arel et al. 2010; LeCun et al. 2015).

Recent developments in computing technology, mainly with graphic processing units (GPU), has accelerated the development of deep neural networks (DNN) for different applications. The seminal work by Hinton et al. (2006) and Bengio et al. (2007) made it possible to train DNNs for complex classification and regression problems with very high accuracy. It is interesting to note that DNNs have been beating all other shallow learning algorithms by huge margins (LeCun et al. 2015) especially in applications such as object recognition (Krizhevsky et al. 2012), image captioning (Vinyals et al. 2015), speech recognition (Graves et al. 2013), video natural language processing (Collobert & Weston 2008), and many more. These considerations make DNNs a very useful tool for the classification of extragalactic radio sources performed in this study.

There are a variety of ways in which radio galaxies can be classified. The classification can be made on a purely morphological basis, or can take other parameters into account, e.g., spectral index, host galaxy brightness at optical/infrared wavelengths, and host galaxy spectra/type. Restricting ourselves to classification schemes based solely on morphology, we find schemes such as the Fanaroff and Riley classification (FR henceforth; Fanaroff & Riley 1974), Wide-angled tailed and Narrow-angled tailed radio galaxies etc. In this study, we have chosen to restrict our investigations to classifications made only on the basis of the radio morphology. The advantage of this choice being that the source samples used are not restricted by the availability of ancillary data. In turn, the classification algorithm is valid for data that have no, or

limited, ancillary data available. This is a major consideration for deep radio surveys as well as surveys that are outside the coverage of available ancillary data.

The first classification scheme that we consider here is the Fanaroff–Riley (FR) classification. The FR scheme divides extended radio galaxies into two classes, designated as FRI and FR II, the membership of which depends on the ratio R of the distance between the brightest points in the source and the total size of the source. Radio galaxies for which $R < 0.5$ are classified as FRI and those for which $R \geq 0.5$ are classified as FR II. Typical features associated with FRI-type radio galaxies include diffuse, plume-like jet(s) and cores, which are brighter than jets/lobes. FR II-type radio galaxies, on the other hand, show bright “hot spots,” typically at the end of the lobes and cores, that are less bright than these. The FR classification scheme, which starts on a morphological basis, also corresponds to a division in radio power ($P_{1.4\text{ GHz}} = 10^{25} \text{ W Hz}^{-1}$) and possibly host galaxy optical luminosity (Ledlow & Owen 1996). For a detailed discussion, see Saripalli (2012).

The FRI/II sources form the bulk of AGN-powered radio galaxies. These are important sources of the feedback processes in the cosmic structure formation (Croton et al. 2006). Several arguments have been put forth to explain the morphological differences, including intrinsic differences in the AGNs powering these sources, the environments of the sources large scale or galactic scales or the mode of the accretion. However, these factors have not been able to successfully explain the FR dichotomy (Gendre et al. 2013). Apart from these sources, there are the so-called FR 0 sources (Sadler et al. 2014; Baldi et al. 2016) as well as sources with “hybrid” morphology (Gopal-Krishna & Wiita 2000), which require further examination. An issue in the latter studies is the relatively small fraction of FRI sources found in current all-sky surveys—due to their relatively high detection and completeness threshold, high redshift and/or low luminosity, low-surface brightness source populations are not probed well. Upcoming all-sky surveys with SKA will probe FRI populations to high redshifts (Kapinska et al. 2015) and would be able to answer these questions (Kharb et al. 2016).

The other category of sources considered in this work is bent-tail sources. Bent-tailed radio galaxies include Wide Angled Tailed (WAT), Head-tail (HT), and Narrow Angled Tailed (NAT) radio galaxies. As their names suggest, these radio galaxies have jets (“tails”) that are bent at an angle from the host optical galaxy. The nature of the angle between the jets determines if the radio galaxy is a WAT or NAT. In some of these galaxies, the jets are swept back to such an extent that they appear as a head (the core) and a tail. These are the HT radio galaxies. The peculiar radio morphology of the bent-tailed sources is generally attributed to their environment, typically a galaxy cluster or a group (Burns 1998). As such, these sources can be used as tracers of clusters of galaxies (Banfield et al. 2016), (Mao et al. 2011), especially at high redshifts, where the information from optical or X-ray bands may be unavailable or sparse.

The plan of the paper is as follows. In the next section, we describe the source sample chosen for training and classification. Section 3 gives a concise background of Convolutional Neural Networks (CNNs). Section 4 contains a description of the specific neural network model we have chosen, the pre-processing needed for the sample source images, and the training process. Section 5 explains the classification model

used to determine the final classification of the sources. Section 6 presents the results and discussion. In Section 7, we briefly summarize the study and present conclusions.

2. Sample Selection

In this section, we describe the sample formation for this study. We have formed separate samples for FRI, FRII, and Bent-tailed radio galaxies respectively. The factors to consider while selecting the samples were high sample numbers and images that are well-resolved as well as freely available. With these constraints, we have decided to restrict ourselves to sources from the Faint Images of Radio Sky at Twenty Centimeter (FIRST; Becker et al. 1995) radio survey. As described below, since there are no source samples of each category that are sufficiently large, we have combined several different samples of sources further, creating artificial sources by processing the sources from these samples (see Section 4.1 for details).

We initially selected the FRI-II sample from a subset of the Combined NVSS and FIRST Galaxies sample (CoNFIG henceforth; Gendre & Wall 2008; Gendre et al. 2010). This sample of radio sources was compiled specifically to address the need and lack of samples of FRI-II sources in the literature. The CoNFIG sample was compiled from an overlapping region from the NRAO VLA Sky Survey (NVSS; Condon et al. 1998) and FIRST surveys. The CoNFIG sample is divided into four sub-samples of varying flux density limits in NVSS, named CoNFIG-1-4, with $S_{1.4\text{ GHz}} \geq 1.3, 0.8, 0.2$ and 0.05 Jy respectively (CoNFIG-2-4 are spatial subsets of CoNFIG-1). It should be noted that even the faintest sources in the sample are bright relative to the bulk of the sources expected to be detected in upcoming surveys.

In total, the source catalog from CoNFIG contains 859 sources. This CoNFIG sample was classified by morphological basis into two categories, FRII and FRI radio galaxies (as well as Compact sources and sources of Uncertain morphology, which we do not include in this study). Because the NVSS images for most of these sources are unresolved with the NVSS beam FWHM of $45''$, the structural information is obtained with FIRST images, which have a beam FWHM of $5''$. The criteria for the classification were the presence of “hot spots” at the edge of radio lobes as well as the alignment of the lobes (if the lobes showed hot spots and were aligned, the source was classified as FRII; collimated jets and hot spots close to the core were taken as signs of FRI radio galaxies—note that this includes bent-tailed radio galaxies). In the present study, we make use of only the sources classified as FRI/II from these. The FRI/II radio galaxies have an associated flag, which can be understood as the degree of confidence in the classification of the source; the flag can either be “confirmed” or “possible.” The final classification of the sample provides 71 FRIs (50 confirmed) and 406 FRIIs (with 390 confirmed). As an initial sample, we have chosen the 50 confirmed FRIs and 390 confirmed FRIIs. The sparsity of the FRI-type radio galaxies is due to the relatively shallow flux density limits of the CoNFIG survey. For example, at $z = 0.15$, the median redshift of FRI radio galaxies in CoNFIG-4 (which is the deepest and spatially the smallest CoNFIG region), the limiting flux density for the other three regions corresponds to a radio power above the nominal radio power divide between FRI/FRII classes. The bulk of the FRIs comes from low redshifts, while the reverse is true for the FRIIs.

To supplement the smaller number of the FRI radio galaxies and address the imbalance in the training set (see Section 4.1 for more details), we decided to include the recent FRICAT catalog of FRI radio galaxies (Capetti et al. 2016). The FRICAT catalog is a subsample of the Best & Heckman (2012) sample, which imposed an upper redshift cut of $z = 0.15$, giving an initial sample of 3357 sources. A further constraint of the size of the radio emission of at least 30 kpc from the center of the host galaxy as seen in FIRST images was applied (corresponding to $11.4''$ for the most distant objects in the FRICAT catalog, thus giving several resolution elements for the smallest source in the sample). Furthermore, sources displaying only FRI morphology (one sided and two sided jets as well as narrow-angled, tailed objects were included). This classification was done visually by all three authors independently and a source was included in the catalog if at least two of the authors agreed on the classification. This makes the classification more robust; this is also similar to the procedure we have adopted independently (see Section 5). Including the FRICAT model gives another 219 FRIs (we have excluded the sample of small FRI galaxies included in the FRICAT catalog in the present study). It should be noted that the majority of the FRI source sample for this study is from the low redshift universe, while the majority of FRII radio galaxies corresponds to relatively high redshifts. This also means that for a given physical extent, FRIs would have more structural detail.

For bent radio galaxies, we have made use of the catalog from Proctor (2011), where the FIRST radio source database has been classified along morphological categories using a combination of pattern-recognition tools and visual inspection (the latter for sources with more than four components and thus expected complex morphology). For details of the classification method, see Proctor (2003) and Proctor (2006). In brief, sources in the FIRST catalog are separated into groups, with low-count groups (those with fewer than three members) being classified using decision tree pattern-recognition techniques in various categories (WAT, NAT, W-shaped sources etc.), and higher-count groups classified using visual inspection. We make use of only the latter category to form a sample of bent-tailed sources. These sources have been visually examined and classified into a variety of types. From these, we have chosen only the confirmed WAT and NAT radio galaxies, excluding those sources where the WAT and NAT identification is uncertain (marked by “?” next to the classification in the table). This gave us 299 bent-tailed radio galaxies. From these initial samples, we have excluded all of the source images in which there are strong artifacts. We have also excluded those images that contain multiple sources, as well as sources too large to fit in the cutout and sources too small to have sufficient structural information. This reduces the sample size to 47 FRIs from the CoNFIG sample and 145 FRIs from the FRICAT sample (giving 192 FRIs in total), 298 FRIIs and 288 bent-tailed radio galaxies. Since all of these are samples based on visual inspection, there is a possibility of confusion in the assigned class of a source due to different studies estimating different morphologies for the same source. To resolve this issue, we have excluded all cross-matched sources from the FRI and FRII samples that have been marked as confirmed WAT/NAT, W-shaped, or Ring (and Ring-lobe) morphologies in either Gendre et al. (2010) or Proctor (2011). We have removed the bent-tailed radio galaxies from the FRICAT by visual inspection. After removing these sources, we are left with 178 FRIs, 284 FRIIs, and 254 bent-tailed radio galaxies. This process is

Table 1
Table Summarizing the Sample Selection Process

	Initial Sample Size	Image- based Cut	Morphology- based Cut	Final Sam- ple Size
FRI Sources	269	77	14	178
FRII Sources	390	92	14	284
Bent-tailed Sources	299	11	34	254

Note. The image-based cut refers to sources excluded due to presence of artifacts, lack of structural information, or very large source size, the morphology-based cut refers to sources discarded due to confusion regarding the morphology.

summarized in Table 1. In the next section, we describe the CNN, which will be trained in classification using this sample of sources.

3. Convolutional Neural Networks

Artificial neural networks (ANN), inspired by biological neurons, try to approximate nonlinear functions from a set of inputs through the combination of simple functions (Cybenko 1989). ANNs generally consist of a network of interconnected neurons, which may have many inputs and a single output like a biological neuron. Having a proper learning rule and activation functions, such interconnected neurons in a specific architecture can be used for classification and regression applications (Jain et al. 1996).

The output y of a single neuron can be mathematically represented as

$$y = \sum_{j=1}^d w_j x_j + w_0 \quad (1)$$

where x_j are the different inputs to the neuron, w_j are the weights to the corresponding inputs, and w_0 is the bias term. The $w_j x_j$ term represents a dot product. The output y is then usually passed through an activation function. Similar to the action potential in a biological neuron, which decides the rate of neuron firing, the activation function in an artificial neuron restricts the neuron output to normalizable values.

$$\hat{y} = f(y), \quad (2)$$

where f is the activation function. The activation functions are of different types, namely threshold functions, piece-wise linear functions, and sigmoid functions (Duda et al. 2012). A similarly large number of neurons can be interconnected with multiple layers of neurons having a distinct activation function (Duda et al. 2012). Therefore, Equation (1) can be rewritten as

$$Net_k = \sum_j^{n_H} y_j w_{kj} + w_{k0}. \quad (3)$$

In Equation (3), k represents each unit in the output layer and n_H represents the number of hidden layers. Combinations of such n_H layers can be used to learn nonlinear functions with backpropagation (Hecht-Nielsen 1989). During the learning process, the inputs, multiplied with their associated weight and bias propagate from the input layer to the output layer through the different hidden layers of neurons. This is commonly referred to as forward pass or forward propagation. At the

output, the error between the calculated output and expected output is estimated and this error is sent back from the output to the input layer to adjust the weights of the neurons. This is called backward pass or backpropagation.

In a CNN, the dot product in Equation (3) is replaced by a convolution operator. Hence, w_j will be a vector instead of a single value as in the case of a normal neural network. w_j is often called a kernel or filter. This facilitates CNNs to directly operate on raw data such as images or time series data as opposed to feature vectors in normal neural networks (LeCun & Bengio 1995).

Yann LeCun for the first time showed the successful application of CNN to digit recognition (LeCun & Bengio 1995). CNNs have been widely used for image classification (Lawrence et al. 1997), speech signal processing (Hinton et al. 2012), and text classification (Collobert & Weston 2008).

CNNs are also referred to as Time Delay Neural Networks because they are generally insensitive to translations of a pattern (Duda et al. 2012). This property is achieved by a method called weight sharing, which constrains backpropagation to generate the same weight values for a similar region in the input space. The input space refers to the data space that is input into the network, for example, images or time series data. Weight sharing is an important property of CNNs, which allows the generation and extraction of translation independent features from the raw data. It can be explained with the following example. Consider a cutout image of an FRI-type galaxy. The galaxy remains FRI even if the same galaxy is at the center of the image cutout or even at any of the corners, provided it is clearly visible. The same is the case when the spatial size of the galaxy within the image changes. There are specific properties that make the galaxy FRI-type irrespective of its position, size, flipping or mirroring, and tilt. The CNN learns features that are shift, translational, and rotational invariant through weight sharing. This simply means that the set of weight values, which represent the FRI galaxy features, are the same irrespective of translational and rotational variations across different samples of the same type. Thus the weights shared for a specific property of a class are shared among different samples. For example, the set of weights that extract features for the two hot spots of the FRII galaxy are the same for any sample of FRII-type galaxy. CNNs also have the same feedforward operations as that of a conventional neural network, enabling the application of similar learning principles.

One of the main advantages of CNNs is that the input to the network is the raw data, or images in this case, rather than feature values designed by astronomers (Krizhevsky et al. 2012). This enables the network to learn and generate a hierarchy of features with minimal information loss (Oquab et al. 2014). Each layer of convolution learns different features. For example, the first few layers learn simple features such as edges and corners. Successive layers combine these elementary features into more complex features to generalize the input data. This succession of feature learning generates a hierarchy of features (Masci et al. 2011). Figure 1 shows a single layer of convolution in a CNN.

CNNs are generally characterized by a third dimension in the network, often called the depth. In the case of image data, different kernels/filters can be learned at once in a given layer as shown in Figure 1. This enables learning of different features of the input data. Each learned kernel adds to the depth dimension of a layer. The general notion of CNNs is that the

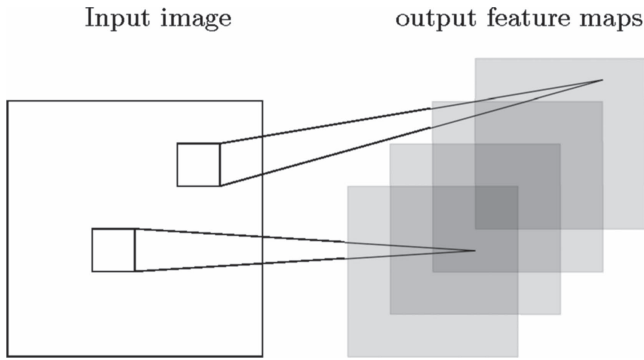


Figure 1. Illustration of a single convolutional layer with multiple output feature maps. With a single input, it is possible to learn different features with different filters/kernels, thus creating a depth of feature maps. The gray squares represent different filters learned in a single layer. The small squares on the input image with pointers to the feature maps simply show the convolution kernel sliding across the image to generate values in the feature maps.

depth increases in the forward direction. The complexity of the features learned in each layer increases in the forward direction and finally they are combined into a fully connected layer. The final layer is usually comprised of a cross-entropy function (Hagenauer et al. 1996; De Boer et al. 2005) for calculating the loss and a scoring layer at the end. The cross-entropy function is a decoding scheme used in information theory, which is based on probability distribution of the sample classes. The final feature layer in a neural network need to be decoded or converted to give the correct class of the training/ test sample. The cross-entropy does this job by looking at the distribution of the feature values. In this study, we will be using a binary cross-entropy function since the models individually will be doing a binary classification.

Deep neural networks are neural networks that have many hidden layers, generally more than two (Hinton et al. 2012). A CNN that has multiple layers of convolutions is termed Deep Convolutional Neural Network (DCNN; Krizhevsky et al. 2012). DCNNs have been widely used for image recognition and speech signal processing applications, and have been performing with exceptional accuracy (Hinton et al. 2012; Krizhevsky et al. 2012; LeCun et al. 2015). In this study, we have made use of DCNN for radio galaxy classification.

Another interesting property of DCNNs is transfer learning (Yosinski et al. 2014). Transfer learning enables us to train a network for a new application with few training examples. In areas like astronomy, it is often difficult to have clean, hand-labeled data sets for different applications. It is possible to exploit the property of transfer learning in DCNNs to train a pre-existing model for a different classification problem. In addition to this, it also improves the existing model without having to retrain it from scratch, as opposed to other, shallow, machine learning methods. One of the main objectives of this study is to also provide a DCNN model that can be used for future transfer learning applications.

DCNNs have been used for different applications in optical astronomy, such as star-galaxy classification (Kim & Brunner 2017) and redshift estimation (Hoyle 2016). In a recent work by Dieleman et al. (2015), a rotational invariant CNN was used for optical galaxy classification, which gave near-human accuracy. These results provide motivation for the application of such techniques to radio astronomy as well.

4. Network Model

The neural network model design, in general, is considered to be a hyperparameter optimization problem. This simply means that there is no strict guideline for the design of a neural network. There is no rule to decide the number of hidden layers or number of neurons for a model. The model design is usually done with respect to the complexity of the data that is investigated. In the case of CNNs, model complexity is generally found to increase with complexity of the objects for classification. Even though simplified models can deliver good accuracies, their prospect for transfer learning (Oquab et al. 2014) are limited. This is because simple models generally have fewer layers of convolutions and activations. For this study, we initially explored different simple models with up to five layers comprised of three to four convolutional layers. During training, these models performed poorly with accuracies below 60%, which is only slightly better than a random guess. One of the objectives of this study is to provide a model that is both complex and standard enough that it can be used for studying more complex source morphologies with fewer training samples enabled by transfer learning. Therefore, we chose to use a standard model that has been successfully used for different transfer learning models (Oquab et al. 2014). We have used a slightly modified version of the Alexnet CNN (Krizhevsky et al. 2012), see Figure 2. This model has been successfully used for different image classification problems and it gave promising results (accuracies greater than 80%) for the initial tests we performed. The advantage of this model is that it can be easily adapted to new classification problems and can also handle background noise in images (Joshi et al. 2012; Sukhbaatar et al. 2014). The original network is designed to work on color images and, in our case, it has been modified to work on single channel images. We have also made corrections to handle the image size and number of classes.

The model consists of five convolutional layers with three sets of maxpooling layers. The maxpooling layers are basically sub-sampling layers. This layer performs down-sampling of an input layer in a nonlinear fashion so as to reduce computational complexity (Boureau et al. 2010) during forward propagation. Each convolution is followed by a Rectified Linear Unit (ReLU; Nair & Hinton 2010), which is a kind of activation layer. This is then followed by a normalization layer (NORM). The final pooled output (Pool5) is then passed onto a series of three fully connected layers, which also have ReLU activations. Figure 3 shows what happens in the main component layers of the network, mainly the convolutional layer and pooling layer.

Within the fully connected layers, neurons having weak connections are dropped out during training. Those neurons/nodes whose weights have a very small value do not interact with other nodes. Such node connections are insensitive to weight updates and are termed as weak connections. These nodes can be discarded off the network since they do not influence the forward pass. This procedure is called *dropout* and is a mechanism to avoid over-fitting (Srivastava et al. 2014). 50% dropout is carried out in this network, which is essentially removing 50% of weak neuron connections.

The fully connected layer FC3 has a depth of two in contrast to the other fully connected layers. The final layer during training calculates the cross-entropy loss (Gold et al. 1996) and during validation outputs a Softmax probability score for the

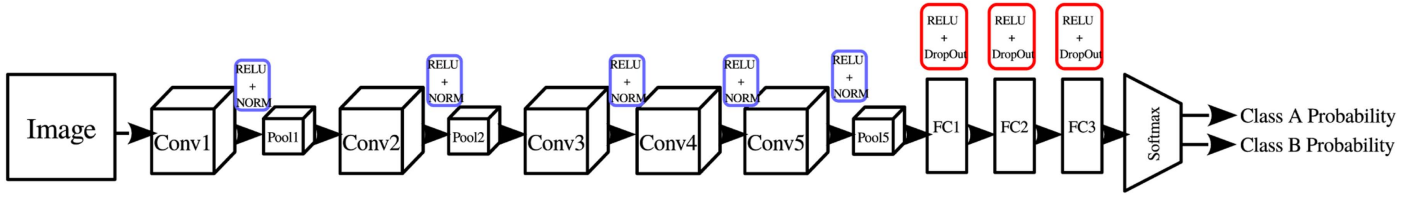


Figure 2. Convolutional neural network architecture used in this study. The model takes in a single channel grayscale image of size 150×150 and outputs the classification score for two classes. The network has a total of 12 layers with 5 convolutional layers, 3 pooling, 3 fully connected, and 1 scoring layer. The scoring layer produces the probability scores for two classes.

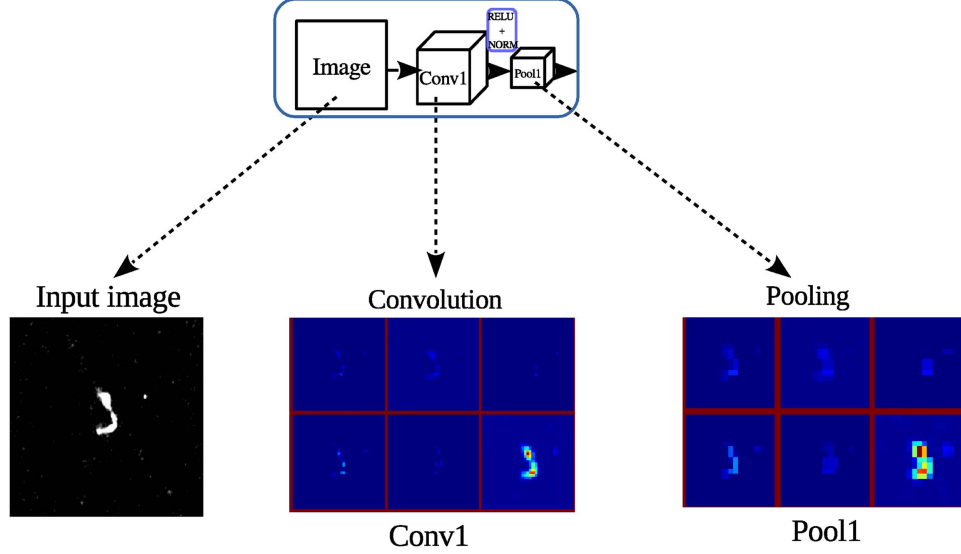


Figure 3. Generalized illustration of the convolution layer and pooling layer of the model. Convolution and pooling layers are the two main components of the DCNN used in this study. The figure shows the output of two such layers in the model. The leftmost column shows a preprocessed input image, which is fed into the network. The second column shows just six filter outputs of the first convolutional layer. It can be seen from the highlighted pixels that each filter/feature map has learned different features of the input image. Not all the filters have learned proper features, which can be seen from the low value pixels in the output. The leftmost column shows outputs of the pooling layer of the corresponding convolutional layers. It can be seen that the pooling layer has down-sampled the output of the first convolutional layer. Similar operations happen in the rest of the network.

two classes. Thus the network has 12 different layers including the different convolutional layers, pooling, fully connected, and softmax layers. The functional description of each layer, kernel sizes, and learned parameters are given in Table 2.

In Table 2, we can see that as the number of convolutional layers increases in the forward direction, and also that the total number of parameters the network has to learn increases dramatically. This is one drawback of DNNs: one needs a large amount of memory to hold these parameters during training. At present, this challenge has been overcome with the advent of GPUs, which can perform fast computations of error gradients in a neural network while also having enough memory to hold a large number of parameters during the training phase. During the testing phase, where there is only forward computation and no backward computation, the memory issue is reduced. Memory overflow can only happen when the batch size of the input is too large or the image size is too big compared to the total available memory.

4.1. Pre-processing Sample Images

Certain image pre-processing steps are desirable before the image is fed into a CNN or any other machine learning algorithms. This is usually done for maintaining the homogeneity of the sample space. This procedure has specific

Table 2
Table of Layer Parameters and Functions

Layer name	Function	Depth	Kernel size	Parameters
Conv1	Convolution	96	11×11	11712
Pool1	Max Pool	96	3×3	
Conv2	Convolution	256	5×5	307456
Pool2	Max Pool	256	3×3	
Conv3	Convolution	384	3×3	307456
Conv4	Convolution	384	3×3	663936
Conv5	Convolution	256	3×3	442624
Pool5	Max Pool	256	3×3	
FC1	Fully Connected Layer	4096		16781312
FC2	Fully Connected Layer	4096		16781312
FC3	Fully Connected Layer	4096×2		8194
Softmax	Softmax Layer	2		
Total Number of parameters learned				35881666

importance with CNNs because the neurons behave like visual receptors similar to human visual receptors. The basic idea is that if a human is able to see an object, the network should also be able to see it. The different stages are shown in Figure 4.

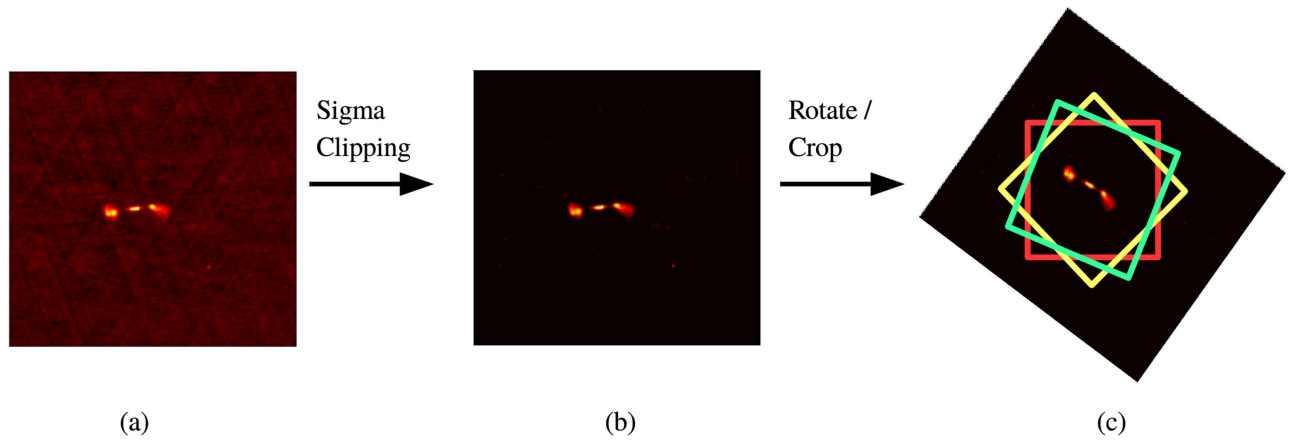


Figure 4. Different stages of image pre-processing before training/testing the network. (a) is the image of the object that is of size 300×300 pixels. (b) The pixels below some noise threshold are suppressed with sigma-clipped statistics. (c) The image is cropped from the center to size 150×150 pixels and similarly for the rotated and flipped versions. The smaller boxes show different cutouts at different rotations.

First, the sigma-clipped statistics⁵ of each image are estimated in order to calculate the background noise and flux levels. With sigma-clipped statistics, pixels above a certain sigma level from the median are discarded or nulled. In this study, all values below a 3σ level of the background were cut-off by suppressing those values to zero so as to highlight the contribution of the source and remove any unwanted background noise. The value for sigma clipping was chosen by training and testing the model at different sigma values. After different iterations and tests, we found that the value of 3σ was better than 2σ or 5σ . In all cases other than 3σ , the model had an accuracy of less than 60%.

One of the important requirements of neural networks is a large number of training samples. With less than 500 training samples in total, it is impossible for the network to learn the features and generalize the different classes. To overcome this issue, training samples can be over-sampled while preserving the labels by rotating and flipping each sample (Krizhevsky et al. 2012). Generally speaking, to train machine learning algorithms, the nominal number of training examples is in the order of 10,000. For DNNs, this number is much larger. In this particular case where we have only a few training samples the label preserving oversampling will help create a large training set to train the network optimally. Here each sample, which is either a rotated or flipped version of a specific image, is treated as a unique training sample. This procedure also helps the network learn rotational invariant features of the samples in the convolutional layers. Another issue that needs to be addressed along these lines is the class imbalance in the synthetic training set. Machine learning algorithms require a fairly equal number of training samples for each class for the model to have good balance between bias and over-fitting (Duda et al. 2012). The model will tend to over-fit if there is a large number of training samples for a specific class compared to the other. In this study, we have balanced the number of training samples by suitably choosing the number of angles for the rotations and their flipping. This procedure will help to produce more evenly distributed samples in the parameter space learned by the convolutional layers.

The different pre-processing steps, shown in Figure 4 can be summarized as follows. The downloaded images were 300×300 pixels in size. The images were rotated by small angles in steps of either 1° , 2° , or 3° such that the number of samples for all of the three classes were roughly equal. The number of rotation steps were chosen such that all the different classes had a roughly equal number of bootstrapped training samples so as to minimize any over-fitting issues in the model. Afterward, a 150×150 patch centered on the source was cut out from the main image. Flipped and rotated versions of the samples were also generated.

4.2. Training

To evaluate machine learning models, all of the data is generally split into two parts. The first part of the data is used to train the machine learning model. The second part of the split data is used to validate the performance of the trained model. This part of the data is known as validation or test data. It is a general practice to take a larger portion of the data to be used for training and the remaining for validation. None of the samples in the validation set are seen by the model during training. In this study, the complete data set for the three classes was split with a 70–30 ratio, where 70% of the original data were taken for training and 30% for validation. So the actual number of training samples were 125 FRIs, 227 FRIIs, and 177 Bent-tailed. The numbers of validation samples were 53, 57, and 77 for FRI, FRII, and Bent-tailed radio galaxies respectively. The data oversampling and augmentation were done for the training set. Thus, the number of samples for FRIs were 45000, 40680 FRIIs and 31860 Bent-tailed “sources.” Afterwards the training data was again split randomly for training and testing with a 80–20 ratio. Therefore the number of samples that went into training from this second split were 36000 FRIs, 32688 FRIIs and 25488 Bent-Tailed “sources.” The training and test samples, being over-sampled versions of the same training images, have overlap since they were generated from the same base samples, but the validation samples which were split from the original 70–30 split were never seen by the network during training.

The network was implemented with the deep learning package called Caffe (Jia et al. 2014), which is widely used in computer vision applications. We used an NVIDIA forked version of Caffe to support training on multiple graphical

⁵ Using the Astropy functionality for sigma clipping, http://docs.astropy.org/en/stable/api/astropy.stats.sigma_clipped_stats.html#astropy.stats.sigma_clipped_stats.

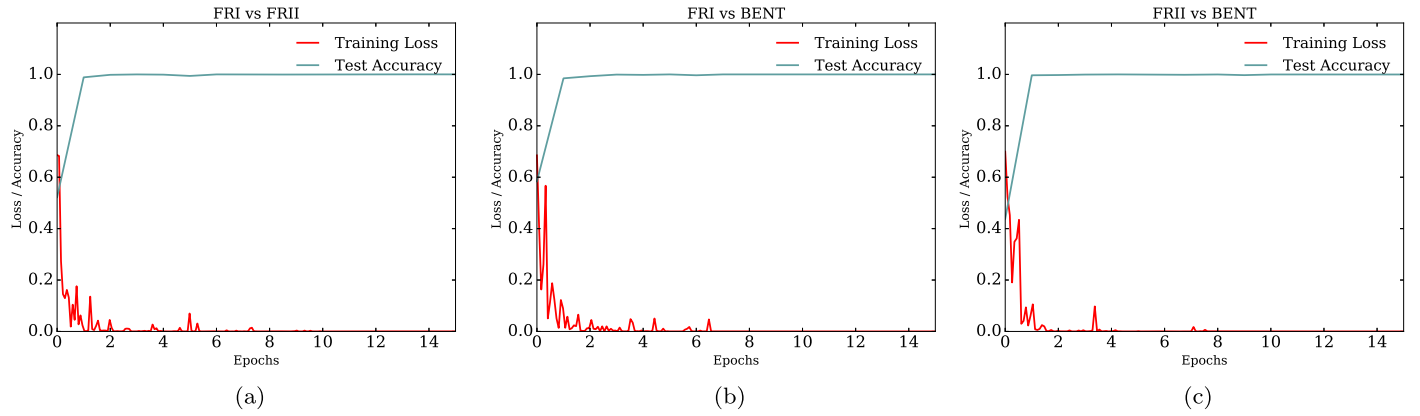


Figure 5. Learning curves showing the training loss and test accuracy for the three different binary classification models. It can be seen that the test accuracy and training loss saturates after 10 epochs for all three models. (a) shows the training and testing accuracy for FRI vs. FRII classification. Similarly (b) and (c) shows the learning curves for FRI vs. bent-tailed and FRII vs. bent-tailed classifications, respectively.

processing unit (GPU) cards. Images in Portable Network Graphics format were converted to a Lightning Memory-Mapped Database for quick access to memory. The training was done on a machine with an Intel(R) Xeon(R) CPU, 260 GB memory and four NVIDIA TITAN-Black GPUs with 12 GB RAM each.

The kernels of each layer were initialized with random Gaussian values. We used a stochastic gradient descent algorithm (Duda et al. 2012) with a batch size of 100 for training. The batch size determines the number of samples that is used for a single forward pass before calculating the backpropagation error by the stochastic gradient descent algorithm. The best learning rate was a step function with a base learning rate of 0.01. The training was done for 30 epochs and a validation of the network was done during every epoch to keep track of the learning performance. The learning curves for the three binary classification models are shown in Figure 5.

The learning curves give a measure of the performance of the machine learning model for the training and testing data (Perlich 2011). The training loss, which is a negative log-likelihood, is calculated from the cross-entropy error (Hinton & Salakhutdinov 2006) and is given as

$$L(w) = -\frac{1}{N} \sum_{n=1}^N [y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)]. \quad (4)$$

In Equation (4), N is the number of training samples, w is the weight vector, \hat{y}_n the expected output and y_n is output during a forward pass. The stochastic gradient algorithm minimizes the error $L(w)$ by properly adjusting the values of the weight vector w . Thus the training loss gives us an idea of how well the model is learning over each iteration or epoch.

The test accuracy is determined when, for each epoch, the model parameters are fixed and no learning takes place, while the model is tested against the test data. The test accuracy computed for each epoch is given as

$$\text{Accuracy} = \frac{1}{N} \sum_{n=1}^N \delta \{\hat{l}_n = l_n\},$$

$$\delta \{\text{condition}\} = \begin{cases} 1 & \text{if condition} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

N being the number of test samples, \hat{l}_n is the predicted class label for the n th sample and l_n is the true label.

With four GPUs, the training for each test case took around 1.2 hr. In all of the cases, we observed that the accuracy tended to saturate to high values after 10 epochs and the training loss fell steeply to very low values. This is because each model is a simple binary classification problem and the network tends to learn quickly without too many training epochs.

4.3. Filter Visualization

Filter visualization of the network model helps us to understand what happens as the learning progresses with each epoch. Different filters learn different properties/features of the object. Looking at the filter visualizations and the learning curves in Figure 5, we can see how the confidence of the network improved with each epoch. During the first few epochs, the network had confusion between the object and the background, and with further learning it gains confidence in distinguishing both. This is shown in Figure 6.

Figure 6 shows the output from two random filters in the first convolutional layer at the first epoch and the last epoch. It is evident that the network has learned to distinguish between the object and the background. Filter 12 has learned the sharp/high frequency features of the radio galaxy and Filter 93 more of the smooth features. In both cases, the network has learned to recognize the radio galaxy in the final epoch. Initially, only a part of the radio galaxy is recognized, while later on the major parts of the galaxy are being recognized. The filter outputs in the initial layers are fairly easy to explain, but as one goes into further layers in the forward directions, the filter visualizations are more difficult to explain (Zeiler & Fergus 2014).

5. Classification Model

With three classes of objects in this study, we trained three different models for binary classification namely FRI versus FRII, FRI versus Bent-tailed radio galaxies, and FRII versus Bent-tailed radio galaxies. Since the actual sample size of the three objects for training is highly imbalanced, there will be a general bias in the models having comparable and large sample numbers. Initially, we trained a single DCNN to classify the three objects together. So the single model would predict if the given sample was either an FRI, FRII, or Bent-tailed radio galaxy. Even though the bootstrapping procedure that generated synthetic training samples to overcome the issues of few training examples and the class imbalance problem, the single

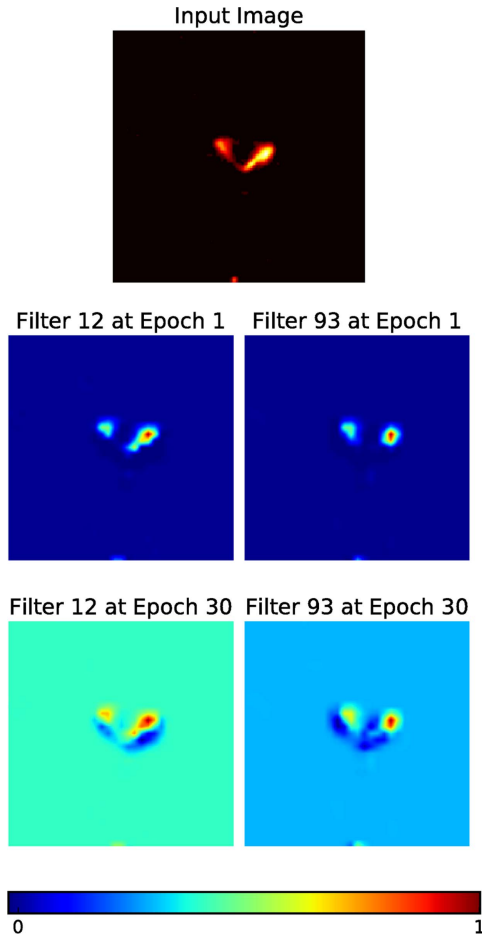


Figure 6. Visualization of the output of two random filters from the first convolutional layer of the network. During the initial epochs the network, the weights had small values (middle row); and, as the learning progressed, the network learned to distinguish between the background and the object. It can be seen from the visualization in Epoch 30 (bottom row) that the weights had solidified with larger values for the object and the background. The filter values here are scaled from their actual values for visualization.

model we trained to classify the three classes performed inefficiently during training and validation. During training, the model showed large training loss, which was a clear indication of poor learning and over-fitting. It was found during training that the training error increased during each epoch and the corresponding test accuracy in each epoch went below 60%. This confirmed that the model was not performing well.

The first solution to minimize this large training loss with a single model was to change the loss function used for training. The Info-Gain loss function is designed and suitable for tackling class imbalance in CNNs (Jia et al. 2014). We experimented the info-gain loss function with different parameters and retrained the network. It was found that with all the different parameters, we tried to optimize with this loss function, the network did not learn the task optimally and both the training and validation results were poor. We then broke the three class classification problem into three binary classifications, which performed better in terms of individual classifications.

To overcome the issue of tuning model complexity, we have made use of a fusion classifier, which is basically a majority voting classifier (Dietterich 2000). This is illustrated in Figure 7.

The fusion model takes the individual predictions of the binary classifiers and their corresponding probabilities to make the final prediction. In general, if for a given sample, two classifiers predict the same class with high probability, then the final class will be the same. However, if the three binary predictions are different and have mixed or low probabilities for their predictions, such samples will be rejected and classified as “strange” objects, and their probability values will be set to zero. This allows users to find objects of potentially interesting or confusing morphology.

6. Results and Discussion

The performance of the fusion model is evaluated on the basis of the classification precision, recall, and F_β score in percentage. The precision gives a measure of correctly classified samples and is given as

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

where TP, true positives, is the number of correctly classified test samples and FP, false positives, is the number of incorrectly classified test samples. The recall which is also called the sensitivity of the classifier is given as

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

where FN is the number of false negatives in the prediction. The recall value can be used to check if the model is over-fitting. For a good model, the precision and recall should be high. The F_β score is a measure that combines both values of precision and recall. The F_β score is expressed as

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

In our test cases, we make use of the F_1 score, where $\beta = 1$. For a good classification, the F_1 score is close to 100%.

The trained model was used to classify 30% validation samples from the FIRST data set using the fusion model. Table 3 shows the classification results for the FIRST samples in the validation set. The “support” column shows the number of test samples in each class. Figure 8 shows some of the predictions by the classification model. The average score for precision, recall, and F1 score are calculated as a weighted average from the receiver operating characteristic (ROC) (Bradley 1997) of the predictions.

From Table 3, we can see that, for all validation samples, the models show excellent precision, recall, and F1 score. The average precision is 88% and the average recall is 86%, with an F1 score of 86%. The results of the fusion classifier can be understood as follows. To be assigned a class, a source needs to be identified as belonging to that class in both the individual classifications which that particular class features.

The bent-tailed radio galaxy classification shows a very high precision at 95%, meaning that most of the classifications labeled as bent-tailed have been identified correctly. The recall for the bent-tailed class is poorer at 79%, which implies that the algorithm was not able to identify all bent-tailed radio galaxies in the validation sample.

The FRI radio galaxy classification shows both high precision and recall—this implies that the network model is able to identify FRI radio galaxies without much confusion.

Table 3

Class of the Source, Size of the Training Samples for Each Class, Precision, Recall, and F1 score of Classification for the Validation Sample As Well As the Support

Class	Training Samples		Precision(%)	Recall(%)	F1 Score(%)	Support
	Actual	Augmented				
Bent-tailed	177	25488	95	79	87	77
FR I	125	36000	91	91	91	53
FR II	227	32688	75	91	83	57
Average			88	86	86	187

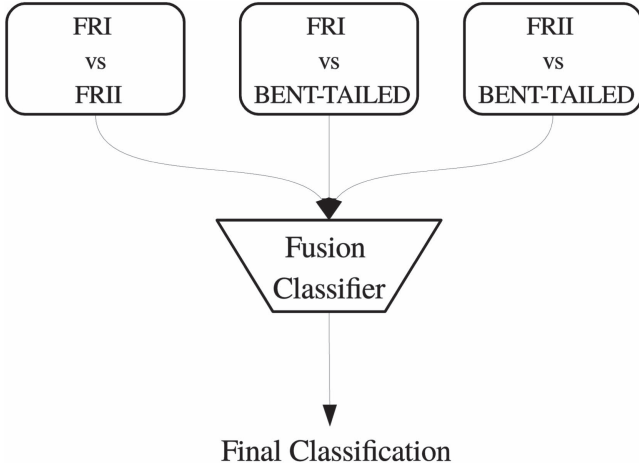


Figure 7. Fusion model with majority voting ensemble classifier, which combines the predictions from the three binary classifier models to make the final prediction. The figure shows the individual predictions from the three binary classifiers being fed into a fusion classifier, which gives the final classification. This model is ideal in situations where the individual models have a slight bias and is also beneficial to identify odd inputs.

The FRII radio galaxy classifications have excellent recall at 91%, but poorer precision at 75% compared to the other two classes. Since the FRI classifications have both high recall and precision, the precision for FRII classification can be directly linked to the recall of bent-tailed sources. This implies that sources that are being identified as FRII are actually bent-tailed radio galaxies. Figure 9 shows these sources for our validation sample, many of these showing two or more bright spots. It may be possible that the algorithm is confused by the bright spots and the diffuse emission in these sources did not get the same “weight,” leading to the misclassification as FRII radio galaxies.

Overall, the results are comparable to manual classification, while being many times faster. This technique, when applied in an iterative manner, would likely reduce the misidentification rate (as seen in FRII-bent classification), with increasing sample size available for training. The effect of training sample size is shown in Table 4. To do this, we chose a training sample that was 25% of the total training sample and created a classification model with it. The validation sample remained the same across the three classification models. The next training sample was obtained by incrementing this sample by a factor of two and generating a classification model with the new training sample. The results show that the average precision, recall, and F1 scores all improve with increasing training sample size.

Table 5 shows some of the predictions with probability for the validation samples with their true class and their coordinates.

One observation that we found during the study was that the CNN was very sensitive to the pre-processing done to the images. During the training of the network, we performed sigma clipping of the images before feeding them to the network. The same procedure has to be done for predictions with the network. Figure 10 shows validation sample J163401.9+062637 before and after pre-processing.

In the example shown in Figure 10, the sample image was incorrectly classified without sigma clipping and was correctly classified with high confidence after the pre-processing. In this case, the actual class label was bent-tail radio galaxy and the prediction without pre-processing was FRII. Depending on the resolution and noise statistics of any image, sigma clipping can have slight effects on the final image, which can also affect the predictions.

7. Wider Application of Deep Learning Model

Machine learning algorithms trained on data from a specific survey have to be retrained to be used on data from other surveys. Shallow machine learning algorithms need to be retrained from scratch for this purpose, which is not realistic in the case of radio astronomy, mainly due to the limitation of sufficient labeled training data. Deep learning methods also suffer from the issue of the need to be retrained; however, DNNs, especially DCNNs like the one in this work, need not be trained from scratch. The idea of transfer learning discussed in Section 3 makes it possible to use an already trained network model to be retrained with fewer examples from a different survey.

The main idea of transfer learning in the context of radio galaxy morphological classification can be explained as follows. The initial layers of the neural network will have learned the basic features like edges and bright spots of the input data. The complicated features are always learned in the last few layers. Thus, in the case of classifying radio galaxies, the initial layers of the network learn the basic shape related features of the different radio galaxies. From Section 4.3, it is evident that the initial layers of the network have learned the basic features of the radio galaxies. For the three different morphologies discussed in this paper, the basic features will be relatively the same irrespective of the survey. The last few layers, which learn more complex features, would be dependent on the resolution and other factors, which differ with surveys. Therefore, it is possible to retrain the network to work on images from other surveys by retraining only the last few layers and freezing the initial layers.

There are different variations and methodologies of transfer learning. The methodologies are designed and optimized for various applications. For different methodologies and applications, the number of training samples required for retraining a network will be different. With some methodologies discussed in Section 3, the required number of training samples needed to

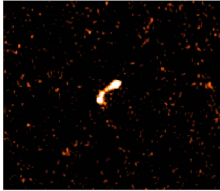
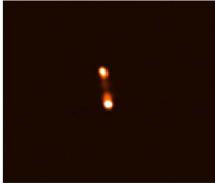
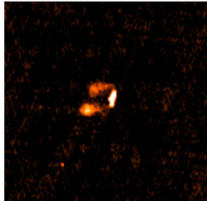
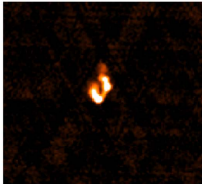
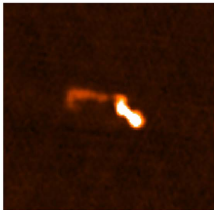
Name, Coordinates	Source Image	Class True / Predicted
J101937.94+001955.7 10 19 37.94 +00 19 55.7		FRI / FRI
3C 228 09 50 10.77 +14 19 57.3		FR II / FR II
J092645.3+030916 09 47 15.887 +52 51 05.20		BENT / BENT
J161637.7+422656 16 16 38.055 +42 27 03.61		BENT / FR II
J013134.5+003341 01 31 34.481 +00 33 33.12		BENT / Strange

Figure 8. Sample predictions made by the classifier model. The first column shows the name of the object and their coordinates, the middle column shows the image cutout, and the left column shows the true class and the predicted class.

be retrained with a new data set is lower compared to the original number of samples that were used to train the model. The number of the training samples required are on the order of thousands for normal image processing applications; however, this has not been tested for any astronomical applications. All applications of transfer learning found in the literature are done with standard imaging data sets specifically designed for computer vision applications with a high signal-to-noise ratio.

Since the signal-to-noise ratio of astronomical images is not comparable to those imaging applications, the numbers associated with the training samples may slightly differ with astronomical images.

A demonstration of the use of transfer learning is beyond the scope of this study. This is because even though the idea of transfer learning is simple, the implementation needs a thorough and systematic analysis because there is no pre-existing study that

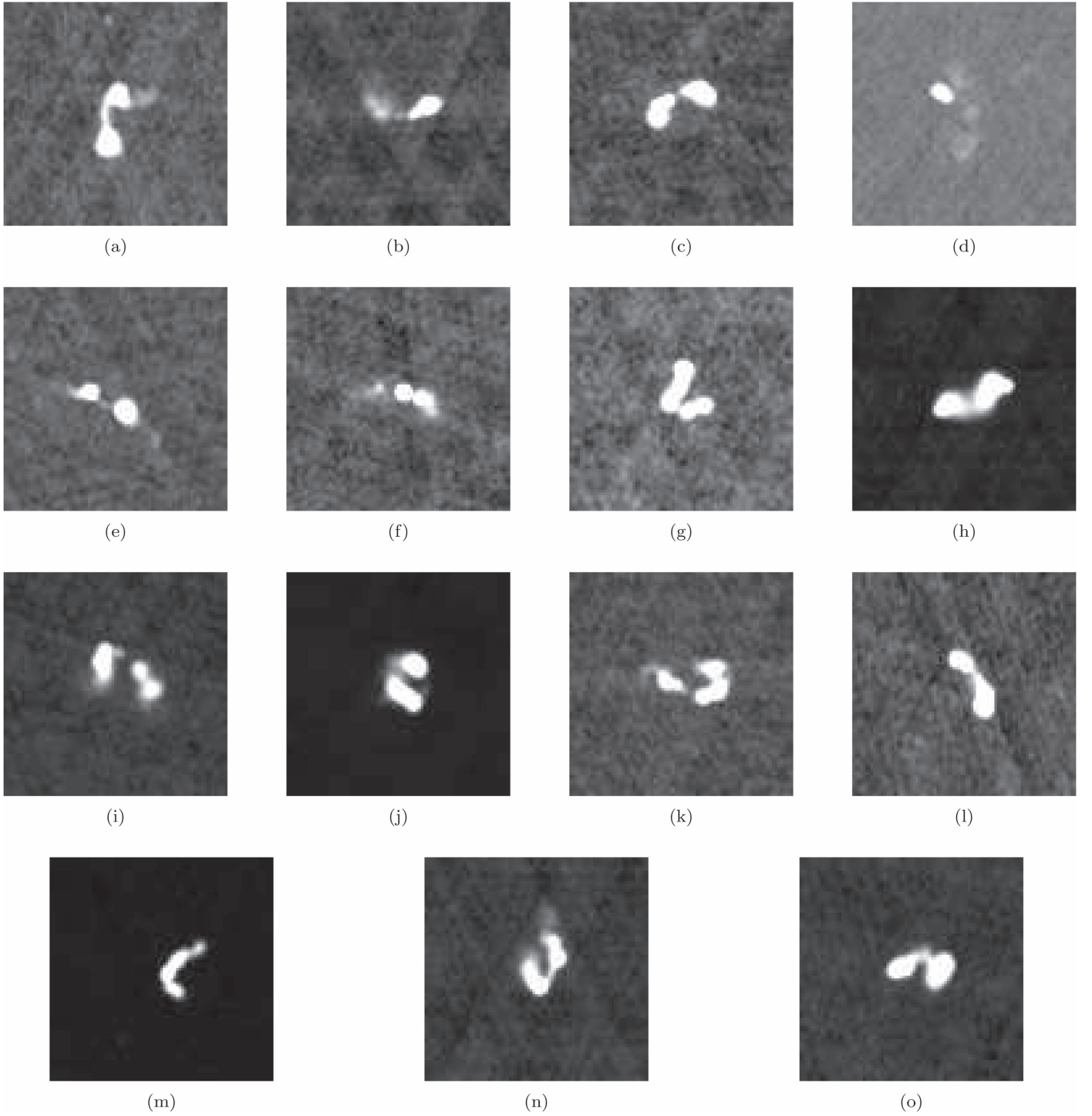


Figure 9. Bent-tailed radio galaxies misidentified as FR-II type radio galaxies. The prediction result for the validation set showed low precision with high recall for FR-II types radio galaxies and high precision with low recall for Bent-tailed radio galaxies. The figure illustrates the effect of this result with many Bent-tailed radio galaxies misclassified as FR-II radio galaxies.

discusses optimizing transfer learning for radio image data. Transfer learning and fine tuning the network for other surveys depend on many factors such as the number of samples in the new data set, selection of layers that need to be retrained, size of the layers, and learning rate. The network may be prone to over-fitting depending on the size of the new data set and content. There is no clear guideline on which of the initial layers should be frozen and optimized specifically for radio images. The assumption with an

already trained network is that it has learned the classification problem with high accuracy (above 90%). Therefore, retraining will be done with smaller learning rates. However, if the accuracy is below certain thresholds, this rule will not hold true. A detailed study on optimizing the implementation of transfer learning for radio images is ongoing and will be published in another paper.

Even though these challenges exist, the model that we present here enables astronomers to use for not only

Table 4
Results of Variation in the Training Sample Size

Relative Sample Size	Avg Precision(%)	Avg Recall(%)	Avg F1 score(%)
25%	54	50	51
50%	66	65	65

Note. The first column shows the training sample size relative to the complete sample described in Table 3 and the other three columns give the respective weighted averages of precision, recall, and F1 score.

classification purposes but also other applications with data from other surveys. With upcoming telescopes, this will enable easy integration of the automated classification system to their science processing pipelines.

8. Conclusions

To summarize, in this study, we demonstrate the utility of Machine Learning Techniques in handling large data sets by using DNNs to classify images of extended radio galaxies. We use archival data from the FIRST radio survey to train as well as test a CNN. Initial samples of ~ 150 – 200 sources were used for each class, augmented by rotated versions of these images to train the network. We test the resulting model on a separate validation sample. The results show that the derived model displays good performance across the source categories, which we have examined. We find that the precision is highest for the bent-tailed radio galaxies, at 95%, whereas it is 91% and 75% respectively, for FRI and FRII classes. The recall is highest for the FRI/II classes at 91% and is at 79% for bent-tailed radio galaxies. These results show that the neural networks can reliably identify different classes of radio galaxies, and are comparable to manual classification, while being much faster, and are thus good techniques for source classification and identification when dealing with large image-based data sets.

At present, deep learning techniques are performing with unprecedented accuracies for different classification problems. Bringing these techniques to radio astronomy is critical for handling the data from upcoming radio facilities such as the SKA and its precursors. Early methods involving pattern recognition and shallow machine learning methods are mainly dependent on handcrafted features, which may not completely capture the properties of the radio galaxies. Our methods with DCNN completely remove the layer of handcrafted feature extraction and builds an end-to-end machine-based model. This method completely embraces the principle of learning from data and is a novel approach in radio astronomy. Another consideration is that of the processing time. The time required to classify a single image with this model is less than 0.17 seconds. Even though the classification is very quick, the inference time for CNN can be further improved with faster GPUs, and by changing the batch size of the input.

Some of the issues we have identified, which pertain to radio astronomy data as well as the specific methods employed, are as follows. One of the main requirements and disadvantages of deep learning models is the large sample size required for training. The level of precision obtained with the present model is mainly dependent on the size of the training samples. Hence, large training samples are essential for the use of “supervised” machine learning methods. Here we have tried to solve the

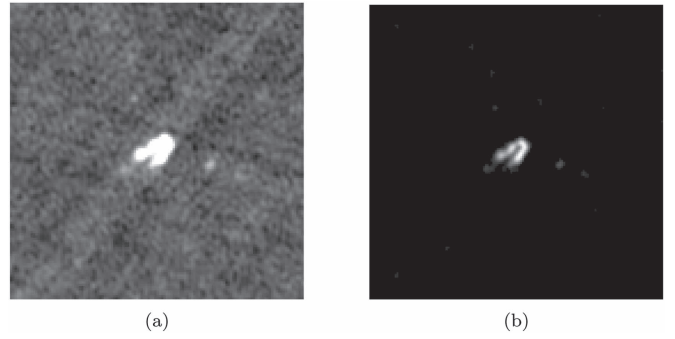


Figure 10. Sample (J163401.9+062637) from the validation set (a) without any pre-processing and (b) after sigma clipping. The sigma-clipped image on the left has far fewer artifacts and background noise compared to the raw image on the right. This shows the effect of pre-processing the images before being fed into the classifier. The sample without the sigma clipping was incorrectly classified during the validation process.

issue by “bootstrapping” the available images to generate a semi-synthetic data set. However, this may result in a smaller feature space for the neural network to run and results for data sets not originating from the same observations may suffer. However, another issue is that the techniques used show heavy dependence on pre-processing. With images from different surveys, the pre-processing will affect the inference of the classifier. Developing machine learning techniques to make inferences in non-stationary environments is still an open problem.

Another issue that affects the quality of the trained model is the number of representative samples that each class originally had. We originally had fewer FRI radio galaxy samples compared to FRII and Bent-tailed radio galaxies. Even though the “bootstrapping” generated enough samples to train the network, the representative samples for each class were different. Therefore the features learned during the training will be confined for each class in the feature space making the model less general and in turn reducing the overall accuracy. We tried to push the accuracy limits of the model by generating the synthetic samples and modifying the loss functions and the success was limited. Since the model allows for transfer learning, this issue can be managed by retraining the model with new samples from future catalogs.

We aim to make the code and model publicly available to the community. The Caffe model and associated code for classification will be available in the public domain at <https://github.com/ratt-ru/toothless>; an archival version has been published to Zenodo (Aniyan & Thorat 2017). An online web service that permits a radio image to be uploaded for classification is also under construction. This will also help improve the model with feedback from the users and by retraining with more samples, enabling the astronomers to use the service for research purposes with better accuracy.

We thank the Square Kilometer Array South African Project (SKA SA), the SKA SA postgraduate bursary program and the South African Research Chair Initiative (SARChI) program for funding the research project. This research has been conducted using resources provided by the Science and Technology Facilities Council (STFC) through the Newton Fund and the SKA Africa. We thank the anonymous referee for the comments and suggestions that have improved the manuscript

Table 5
Table of Predictions for Validation Samples

Source	R.A. h:m:s	Decl. d:m:s	True Class	Prediction	Probability
1426+0093	14 26 49.84	+00 55 59.9	FR II	FR II	99.99995
3C 194	08 10 03.67	+42 28 04.0	FR II	FR II	99.99995
3C 208	08 53 08.83	+13 52 55.3	FR II	FR II	99.99735
3C 228	09 50 10.77	+14 19 57.3	FR II	FR II	99.9999
3C 240	10 17 49.77	+27 32 07.7	FR II	FR II	99.99785
3C 243	10 26 31.96	+06 27 32.7	FR II	FR II	100.0
3C 244.1	10 33 33.87	+58 14 37.9	FR II	FR II	99.9998
3C 251	11 08 37.60	+38 58 42.1	FR II	FR II	99.9981
3C 268.2	12 00 59.77	+31 33 57.9	FR II	FR II	99.996
3C 268.4	12 09 13.52	+43 39 18.7	FR II	FR II	99.99425
3C 277.2	12 53 32.70	+15 42 27.3	FR II	FR II	99.98255
3C 294	14 06 44.10	+34 11 26.2	FR II	FR II	99.99975
3C 322	15 35 01.27	+55 36 49.8	FR II	FR II	99.99985
3C 323.1	15 47 44.23	+20 52 41.0	FR II	FR II	99.9888
3C 336	16 24 39.42	+23 45 17.5	FR II	FR II	99.99995
3C 342	16 36 37.38	+26 48 06.6	FR II	FR II	99.9957
4C -00.55	14 23 26.70	-00 49 56.5	FR II	FR II	99.99925
4C 01.39	13 57 01.51	+01 04 39.7	FR II	FR II	99.9999
4C 03.21	11 11 22.71	+03 09 10.4	FR II	FR II	86.14455
4C 05.53	11 48 47.51	+04 55 27.7	FR II	FR II	99.99995
J151056.2+054441	15 10 55.851	+05 44 39.29	BT	FR II	99.95345
J151744.96+310015.8	15 17 44.96	+31 00 15.8	FRI	FRI	99.9999
J152439.9+620225	15 24 42.006	+62 02 50.93	BT	BT	99.9971
J152522.33+314037.1	15 25 22.33	+31 40 37.1	FRI	FRI	99.99995
J153522.1+342247	15 35 22.994	+34 23 02.98	BT	BT	99.9821
J153616.2+142045	15 36 16.805	+14 20 41.16	BT	BT	99.6882
J153932.09+013710.5	15 39 32.09	+01 37 10.5	FRI	FRI	100.0
J154549.4-024954	15 45 48.671	-02 49 59.76	BT	BT	99.99845
J155222.36+223311.9	15 52 22.36	+22 33 11.9	FRI	FRI	99.9942
J155721.38+544015.9	15 57 21.38	+54 40 15.9	FRI	FRI	99.9996
J160318.6+192414	16 03 18.856	+19 24 18.13	BT	BT	99.97035

(This table is available in its entirety in machine-readable form.)

considerably. We would also like to thank Prof. Oleg Smirnov, Dr. Julien Girard, Spheisihle Makhathini, Etienne Bonnassieux, Dr. Nadeem Oozeer, and Dr. Jasper Horrell for their valuable suggestions and comments. We also thank Dr. Lindsay Magnus for his inputs on the MeerKAT data rates. The authors would also like to thank Dr. Roger Deane for detailed feedback, which was instrumental to this work.

References

- Aniyan, A., & Thorat, K. 2017, ratt-ru/toothless: First release of toothless, v1.0.0, Zenodo, doi:[10.5281/zenodo.579637](https://doi.org/10.5281/zenodo.579637)
- Arel, I., Rose, D. C., & Karnowski, T. P. 2010, *IEEE Computational Intelligence Magazine*, 5, 13
- Baldi, R. D., Capetti, A., & Giovannini, G. 2016, *AN*, 337, 114
- Banfield, J. K., Andernach, H., Kapińska, A. D., et al. 2016, *MNRAS*, 460, 2376
- Banfield, J. K., Wong, O. I., Willett, K. W., et al. 2015, *MNRAS*, 453, 2326
- Bates, S. D., Bailes, M., Barsdell, B. R., et al. 2012, *MNRAS*, 427, 1052
- Becker, R. H., White, R. L., & Helfand, D. J. 1995, *ApJ*, 450, 559
- Bengio, Y. 2009, *Foundations and trends in Machine Learning*, 2, 1
- Bengio, Y., & LeCun, Y. 2007, *Large Scale Kernel Machines* (Cambridge, MA: MIT Press)
- Benitez, N. 2000, *ApJ*, 536, 571
- Best, P. N., & Heckman, T. M. 2012, *MNRAS*, 421, 1569
- Blum, A. L., & Langley, P. 1997, *Artificial Intelligence*, 97, 245
- Boureau, Y.-L., Ponce, J., & LeCun, Y. 2010, in *Proc. 27th Int. Conf. on Machine Learning (ICML-10)* (Madison, WI: Omnipress), 111
- Bradley, A. P. 1997, *Pattern Recognition*, 30, 1145
- Burns, J. O. 1998, *Sci*, 280, 400
- Capetti, A., Massaro, F., & Baldi, R. D. 2016, arXiv:[1610.09376](https://arxiv.org/abs/1610.09376)
- Cavuoti, S., Amaro, V., Brescia, M., et al. 2017a, *MNRAS*, 465, 1959
- Cavuoti, S., Tortora, C., Brescia, M., et al. 2017b, arXiv:[1701.08120](https://arxiv.org/abs/1701.08120)
- Chen, H. 1995, *Journal of the American Society for Information Science*, 46, 194
- Collobert, R., & Weston, J. 2008, in *Proceedings of the 25th International conference on Machine learning*, ACM (New York, NY: ACM), 160
- Condon, J. J., Cotton, W. D., Greisen, E. W., et al. 1998, *AJ*, 115, 1693
- Croton, D. J., Springel, V., White, S. D. M., et al. 2006, *MNRAS*, 365, 11
- Cybenko, G. 1989, *Mathematics of Control, Signals and Systems*, 2, 303
- De Boer, P.-T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. 2005, *Annals of operations research*, 134, 19
- Dieleman, S., Willett, K. W., & Dambre, J. 2015, *MNRAS*, 450, 1441
- Dietterich, T. G. 2000, in *Int. Workshop on Multiple Classifier Systems*, Lecture Notes in Computer Science (Berlin: Springer), 1
- Duda, R. O., Hart, P. E., & Stork, D. G. 2012, *Pattern Classification* (New York: Wiley)
- Eatough, R. P., Molkenhuth, N., Kramer, M., et al. 2010, *MNRAS*, 407, 2443
- Fanaroff, B. L., & Riley, J. M. 1974, *MNRAS*, 167, 31P
- 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, *MNRAS*, 430, 3086
- Gendre, M. A., & Wall, J. V. 2008, *MNRAS*, 390, 819
- Gold, S., & Rangarajan, A. 1996, *Journal of Artificial Neural Networks*, 2, 381
- Gopal-Krishna, & Wiita, P. J. 2000, *A&A*, 363, 507
- Graves, A., Mohamed, A.-r., & Hinton, G. 2013, in *IEEE Int. Conference on Acoustics, Speech and Signal Processing*, 6645
- Guyon, I., & Elisseeff, A. 2006, *Feature Extraction* (Berlin: Springer) 1–25
- Hagenauer, J., Offer, E., & Papke, L. 1996, *ITIT*, 42, 429
- Hecht-Nielsen, R. 1989, in *Int. Joint Conf. on Neural Networks, IJCNN (IEEE)*, 593
- Hinton, G., Deng, L., Yu, D., et al. 2012, *ISPM*, 29, 82
- Hinton, G. E., Osindero, S., & Teh, Y.-W. 2006, *Neural Computation*, 18, 1527
- Hinton, G. E., & Salakhutdinov, R. R. 2006, *Sci*, 313, 504

- Hocking, A., Geach, J. E., Davey, N., & Sun, Y. 2015, arXiv:1507.01589
- Hoyle, B. 2016, *A&C*, **16**, 34
- Jain, A. K., Mao, J., & Mohiuddin, K. M. 1996, *IEEE Computer*, **29**, 31
- Jia, Y., Shelhamer, E., Donahue, J., et al. 2014, in Proc. 22nd ACM Int. Conf. on Multimedia (New York, NY: ACM), 675
- Joshi, A. J., Porikli, F., & Papanikolopoulos, N. P. 2012, *ITPAM*, **34**, 2259
- Kapinska, A. D., Hardcastle, M., Jackson, C., et al. 2015, *Advancing Astrophysics with the Square Kilometre Array (AASKA14)*, **173**
- Kharb, P., Lal, D. V., Singh, V., et al. 2016, *JApA*, **37**, 34
- Kim, E. J., & Brunner, R. J. 2017, *MNRAS*, **464**, 4463
- Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. 2007, *Supervised Machine Learning: A Review of Classification Techniques* (Amsterdam: IOS Press)
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, *Advances in Neural Information Processing Systems* (Cambridge, MA: MIT Press), 1097
- Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. 1997, *ITNN*, **8**, 98
- LeCun, Y., & Bengio, Y. 1995, *The Handbook of Brain Theory and Neural Networks* (Cambridge, MA: MIT Press), 3361
- LeCun, Y., Bengio, Y., & Hinton, G. 2015, *Natur*, **521**, 436
- Ledlow, M. J., & Owen, F. N. 1996, *AJ*, **112**, 9
- Mahabal, A., Djorgovski, S., Drake, A., et al. 2011, arXiv:1111.0313
- Mahabal, A., Djorgovski, S., Williams, R., et al. 2008, in AIP Conf. Proc. 1082, *Classification and Discovery in Large Astronomical Surveys*, ed. C.A.L. Bailer-Jones (Melville, NY: AIP), 287
- Mao, M. Y., Sharp, R., Saikia, D. J., et al. 2011, *JApA*, **32**, 585
- Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. 2011, in Int. Conf. on Artificial Neural Networks, *Lecture Notes in Computer Science*, Vol 6791, ed. T. Honkela (Berlin: Springer), 52
- Morello, V., Barr, E. D., Bailes, M., et al. 2014, *MNRAS*, **443**, 1651
- Nair, V., & Hinton, G. E. 2010, in Proc. 27th Int. Conf. on Machine Learning (ICML-10) (Madison, WI: Omnipress), 807
- Norris, R. P., Hopkins, A. M., Afonso, J., et al. 2011, *PASA*, **28**, 215
- Oquab, M., Bottou, L., Laptev, I., & Sivic, J. 2014, in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1717
- Padovani, P. 2016, arXiv:1609.00499
- Perlich, C. 2011, *Encyclopedia of Machine Learning* (Berlin: Springer), 577
- Polsterer, K. L., Gieseke, F., & Igel, C. 2015, in ASP Conf. Ser. 495, *Astronomical Data Analysis Software and Systems XXIV*, ed. A. R. Taylor & E. Rosolowsky (San Francisco, CA: ASP), 81
- Proctor, D. D. 2003, *JEI*, **12**, 398
- Proctor, D. D. 2006, *ApJS*, **165**, 95
- Proctor, D. D. 2011, *ApJS*, **194**, 31
- Sadler, E. M., Ekers, R. D., Mahony, E. K., Mauch, T., & Murphy, T. 2014, *MNRAS*, **438**, 796
- Saripalli, L. 2012, *AJ*, **144**, 85
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, *Journal of Machine Learning Research*, **15**, 1929
- Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., & Fergus, R. 2014, arXiv:1406.2080
- van Velzen, S., Falcke, H., & Körding, E. 2015, *MNRAS*, **446**, 2985
- Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. 2015, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 3156
- Wagstaff, K. L., Tang, B., Thompson, D. R., et al. 2016, *PASP*, **128**, 084503
- Weir, N., Fayyad, U. M., & Djorgovski, S. 1995, *AJ*, **109**, 2401
- Wilman, R. J., Jarvis, M. J., Mauch, T., Rawlings, S., & Hickey, S. 2010, *MNRAS*, **405**, 447
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. 2014, *Advances in Neural Information Processing Systems*, 3320
- Zeiler, M. D., & Fergus, R. 2014, in European Conference on Computer Vision (Berlin: Springer), 818