Morphological classification of radio galaxies: Capsule Networks versus Convolutional Neural Networks

Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **487,** 1729–1744 (2019)
Advance Access publication 2019 May 20

doi:10.1093/mnras/stz1289

Morphological classification of radio galaxies: capsule networks versus convolutional neural networks

V. Lukic,[1⋆] M. Brüggen,[1⋆] B. Mingo,[2] J. H. Croston,[2] G. Kasieczka[3] and P. N. Best[4]

[1] *Hamburg Observatory, University of Hamburg, Gojenbergsweg 112, D-21029 Hamburg, Germany*
[2] *School of Physical Sciences, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK*
[3] *Institute of Experimental Physics, University of Hamburg, Luruper Chaussee 149, D-22761 Hamburg, Germany*
[4] *Institute for Astronomy, University of Edinburgh, Royal Observatory, Blackford Hill, Edinburgh EH9 3HJ, UK*

Accepted 2019 May 5. Received 2019 April 25; in original form 2018 October 18

ABSTRACT

Next-generation radio surveys will yield an unprecedented amount of data, warranting analysis by use of machine learning techniques. Convolutional neural networks are the deep learning technique that has proven to be the most successful in classifying image data. Capsule networks are a more recently developed technique that use capsules comprised of groups of neurons that describe properties of an image including the relative spatial locations of features. This work explores the performance of different capsule network architectures against simpler convolutional neural network architectures, in reproducing the classifications into the classes of unresolved, FRI, and FRII morphologies. We utilize images from a LOFAR survey which is the deepest, wide-area radio survey to date, revealing more complex radio-source structures compared to previous surveys, presenting further challenges for machine learning algorithms. The four- and eight-layer convolutional networks attain an average precision of 93.3 per cent and 94.3 per cent, respectively, compared to 89.7 per cent obtained with the capsule network, when training on original and augmented images. Implementing transfer learning achieves a precision of 94.4 per cent, which is within the confidence interval of the eight-layer convolutional network. The convolutional networks always outperform any variation of the capsule network, as they prove to be more robust to the presence of noise in images. The use of pooling appears to allow more freedom for the intra-class variability of radio galaxy morphologies, as well as reducing the impact of noise.

Key words: instrumentation: miscellaneous – methods: miscellaneous – methods: data analysis, surveys – radio continuum: galaxies – radio continuum: general.

1 INTRODUCTION

Active Galactic Nuclei (AGNs) are energetic, astrophysical sources powered by accretion on to supermassive black holes in galaxies (Fabian 1999; Padovani 2017). There are many classes of AGNs, where one subset is radio-loud AGN, also known as radio galaxies. The two main ways of classifying radio galaxies is by the properties of optical emission lines (Hine & Longair 1979) or by the radio morphology of the jets (Bicknell 1995). The classification of radio galaxy morphology is of research interest in wide-field radio surveys as it correlates with physical properties of the galaxy such as the total power, dust distribution, surrounding environment, and galaxy and cluster evolution (Saripalli 2012). Radio galaxies can present compact or extended radio morphologies (Miraghaei & Best 2017) and are often classified into either the FRI (core-bright) or FRII (edge-bright) galaxies (Fanaroff & Riley 1974). Rarer are hybrid galaxies, which fall in between FRI and FRII galaxies (Gopal & Wiita 2000). There are physical differences between the two classes. The jets of FRIs are less powerful, and are disrupted quite close to the core of the radio galaxy, while the jets of FRII are more powerful and stay relativistic for much larger distances, terminating in a shock (Contopoulos, Gabuzda & Kylafis 2015). The transition from FRII to FRI radio galaxies is thought to occur as the jet becomes sub-relativistic (Bicknell 1994). As the environment plays a large role in the morphology of radio galaxies, it is not unusual for both lobes to have different appearances, especially the FRIs. The dynamics of the ambient gas and the motion of the host galaxy can create tails or distort the jets through ram pressure stripping (Feretti 2003). Compact radio sources may be either scaled-down (young) versions of the FRI or FRII sources, or may represent a physically distinct population (Baldi, Capetti & Giovannini 2015).

⋆ E-mail: vesna.lukic@hs.uni-hamburg.de (Vl); mbrueggen@hs.uni-hamburg.de (MB)

Radio surveys map ever-increasing numbers of radio sources. The visual classification of such sources becomes increasingly time consuming and will be completely unfeasible with the rapidly increasing data volumes. Recent and upcoming surveys, such as the LOFAR Two-Metre Sky Survey (LoTSS; Shimwell et al. 2017), the Evolutionary Map of the Universe (EMU; Norris et al. 2011), and surveys with the Square Kilometre Array (SKA; Prandoni & Seymour 2015) will detect many millions of galaxies. Citizen science projects have been used for classifying astronomical sources, for example in Galaxy Zoo 2 (Willett et al. 2013, Dieleman et al. 2015) and Radio Galaxy Zoo (RGZ, Banfield et al. 2015). It is also possible to use automated techniques to classify images. Ultimately, these approaches can be used as a training set for machine learning algorithms, in particular deep learning algorithms, when the data are high-dimensional (Wu et al. 2018).

The most prominent wide-area radio surveys, such as the Faint Images of the Radio Sky at Twenty centimetres (FIRST; Becker, White & Helfand 1995) and the NRAO VLA Sky Survey (NVSS; Condon et al. 1998), have mostly been conducted at GHz frequencies. In contrast, the LoTSS survey, which is the focus of this work, has been carried out at 150 MHz with the Low-Frequency Array (LOFAR). As such, LOFAR can detect synchrotron emission from older populations of relativistic electrons (which have steeper spectra) found in the extended regions of sources. Furthermore, with its combination of long and short baselines, LoTSS offers both a high angular resolution ($\approx$6 arcsec) for detailed mapping, and a high sensitivity to extended emission.

The cross-identification of radio sources with their optical or infrared hosts helps to associate radio components to sources and to determine properties, such as host galaxy redshift and mass. Previously, cross-identification has been done using visual input from citizen scientists input in RGZ (Banfield et al. 2015), and automated methods in cross-identifying radio emission with infrared counterparts have been explored (Alger et al. 2018). In the LoTSS survey (Shimwell et al. 2019) the radio sources have been cross-matched with their optical counterparts. For the majority of sources a maximum-likelihood ratio test was adequate because the sources are small and unresolved. For sources that are too large or complex, a visual host identification has been applied (Williams et al. 2019).

The first published work on the automated image classification of radio sources using deep learning algorithms was Aniyan & Thorat (2017) where they use a limited number of original radio galaxy images and apply aggressive augmentation to classify sources into FRI, FRII, and bent-tailed classes. In previous work, we have shown that it is possible to classify radio sources into four categories based on the number of components belonging to the radio source and produced a classification accuracy of 94.8 per cent (Lukic et al. 2018) on the RGZ DR1 catalogue (Wong et al, in preparation). Alhassan, Taylor & Vaccari (2018) developed a convolutional neural network model to classify FIRST sources into four classes including compact, FRI, FRII, and bent-tailed sources, achieving overall accuracies >90 per cent. Wu et al. (2018) use regional convolutional networks to localize, recognise, and classify sources, the best model obtaining a final mean average precision of 83.4 per cent, using the number of peaks and number of components of a particular radio source. This approach, however, does not always lend itself easily to clear morphological classifications in the FRI or FRII cases because the relative orientations of components are not taken into account.

The aim of this work is to compare the performance of two set-ups of deep learning networks (capsule networks and convolutional networks) in the classification of radio sources. As a data set, we used the first data release of the LoTSS survey (Shimwell et al. 2019). Capsule networks are a more recently developed deep learning technique, invented to help preserve the local feature information within an image, which can be degraded in traditional convolutional networks, owing to the pooling operation. In the context of radio galaxies, the orientation and pattern of the emission is important as it determines the morphological classification. The data from the LOFAR LoTSS survey reveals sources in unprecedented detail, therefore one source that had a particular morphology in an earlier survey may be revealed to have a different one when imaged with LOFAR.

This paper is outlined as follows: Section 2 describes the LOFAR data set, including catalogue information and image data as well as how the classifications are generated. Section 3 discusses the pre-processing and augmentation applied to the original images. Section 4 describes the theory behind the two deep learning approaches explored, namely convolutional neural networks and capsule networks. Section 5 explores the performance of different capsule network models against standard convolutional neural network set-ups, including transfer learning on the LOFAR data, when training on different sets of images. The results are also discussed in Section 5. Section 6 summarizes our overall findings.

2 LOFAR HETDEX V1.0 DATA SET

2.1 Source cut-outs

The sources in our data set originate from a 424 square degree region of the HETDEX Spring Field, mapped from the LoTSS, and release as Data Release 1 (Shimwell et al. 2019). The LoTSS survey detects a total of 325 694 sources where the signal is five times that of the noise and the density of sources is a factor of approximately 10 times higher than the most sensitive existing very wide-area radio-continuum surveys. We use v1.0 of the value-added catalogue for the HETDEX-area data release of LoTSS. The first step in creating the value-added catalogue involved using PyBDSF[1] to produce a radio source catalogue for the field, after which a decision tree was used to further categorize the sources, with details provided in Williams et al. (2019). After filtering the 325 694 sources to only include those classified as resolved leaves 24 096 sources (Shimwell et al. 2019). The catalogue also contains 180 columns describing the properties, such as redshift, position etc, of the sources. In order to exclude star-forming galaxies and sources with less certain redshift values, we used the AGN subsample of the LoTSS catalogue, derived by Hardcastle et al. (2019) leaving 6708 sources. We note that this is a substantial limitation of the machine learning approach when using radio galaxy image data only, as it is generally not always possible to filter out the star-forming galaxies without the use of additional data at other wavelengths. The source classifications were only available for those 6708 sources classified as AGNs and with known redshifts, therefore the analysis is restricted to this set. However, the accurate knowledge of redshift is not strictly required for morphological classification.

Finally, we assume that there is one source per image. Square cut-outs of each source are produced from the FITS images, where the cut-out size is determined by the catalogued size of the radio source. These range from size (66,66) pixels up to (2342,2342) pixels. The size of the pixels is roughly 1.5 $\times$ 1.5 arcsec. Fig. 1

[1] http://www.astron.nl/citt/pybdsf/.

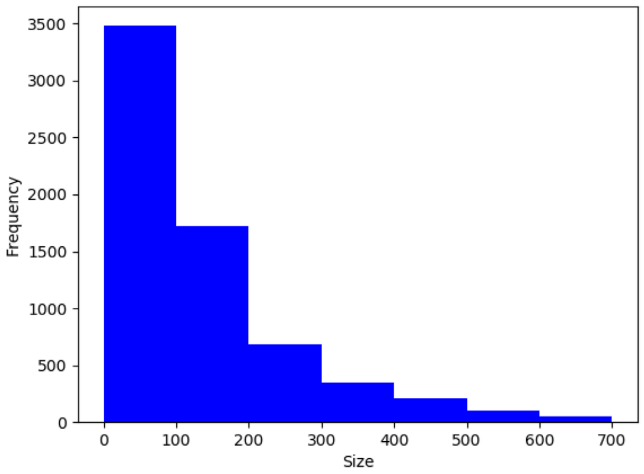

Figure 1. Histogram of sizes (in pixels per side) of the filtered cut-out images. The total number of images is 6708.

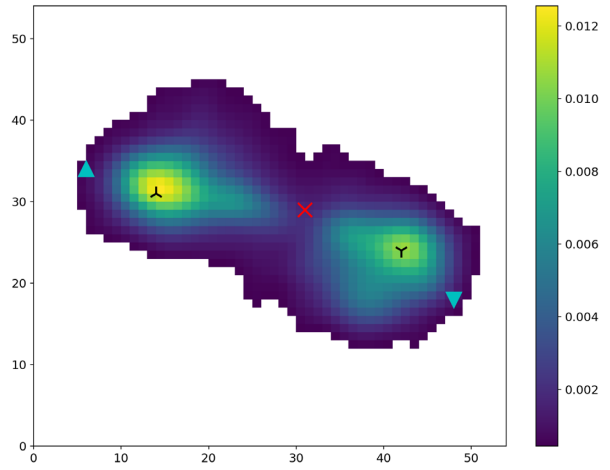

Figure 2. The masked array from which classifications are generated. The red cross indicates the position of the optical source, the black Y's indicate the peaks of the emission, and the blue triangles indicate the maximum extents of emission. The optical position is calculated from the user's clicks on the LoTSS images, or from the maximum likelihood method. The Y's and blue triangles are outputs from the automated classification code.

shows the histogram of the side length in pixels of the images for these 6708 samples.

2.2 Classifications

The LoTSS association and cross-identification effort (Williams et al. 2019) was a project in which expert astronomers were tasked with characterizing the radio emission for sources larger than 15 arcsec . Indicated were the locations of the peaks and extents of the emission, and whether there was one or more sources present.

The 6708 source sample (see Section 2.1) were classified into six classes using an automated technique (Mingo et al. in preparation). The six classes are Unresolved-1, FRI, FRII, Hybrid-1, Hybrid-2, and Unresolved-2, all of which are described in further detail as follows. After the host galaxy location had been identified through the LoTSS identification effort (Williams et al. 2019), the distances, d1 and d2, were determined as the distances in pixels from the host galaxy to the brightest peaks of emission on both sides of the source (shown with points marked with Y/inverted Y in Fig. 2). Similarly, Maxd1 and Maxd2 were determined as the maximum extents of the source in each direction (marked with triangles on the plots), out to the masked 4rms limit. A 120 deg aperture cone is used to find those along the direction of d1, d2. The comparison of d1/Maxd1 and d2/Maxd2 is then used to classify the sources. If, on both sides, the peak is less than half of the distance between the position of the host galaxy and the maximum extent of the emission (i.e. d1/Maxd1 $<$ 0.5 and d2/Maxd2 $<$ 0.5) then the source is classified as an FRI, making up 15 per cent of the total sources. Likewise, if it is more than half of the distance (d1/Maxd1 $>$ 0.5 and d2/Maxd2 $>$ 0.5) then the source is classed as an FRII. The FRIIs make up 7 per cent of the total sources.

In addition to the FRI and FRII labels, four further labels were defined. Hybrid-1 and Hybrid-2 classes refer to sources which show FRI morphology on one side of the source and FRII on the other, with the '1' or '2' reflecting the classification of the brighter of the two sides. The *Hybrid* classes together make up 6 per cent of the sources. Unresolved-1 sources correspond to those images that have less than 5 pixels of signal above 4rms, making up 22 per cent of the sources. This class is useful as it indicates which images are too noisy to be characterized into a particular class (note that it is different from the Unresolved sources previously discussed, which were based on the extent of the overall radio emission). Finally, the Unresolved-2 class contains a collection of mostly FRI and FRII sources that were unable to be classified accurately by the automated algorithm as they were too small, which makes up 50 per cent of the sources. Fig. 2 shows an example image source, demonstrating how the classification labels were generated.

Table 1. The number of original and augmented sources, divided into training and testing sets. The percentage of samples in each class is also given for the test set. Since only original images should be used in the test set, the augmented images are used for training only.

Class	# Orig. (Train)	# Orig. (Test)	# Aug.	# Total
Unres.	1156	301 (50.2 %)	4371	5828
FRI	765	219 (36.5 %)	5904	6888
FRII	380	80 (13.3 %)	2760	3220
Total	2301	600	13 035	15 936

In this work, we have chosen the Unresolved-1 (henceforth called Unresolved), FRI, and FRII classes to evaluate the performance of our deep learning algorithms, as these had the most confident classifications. There are 2901 original images in total, as shown in Table 1.

The automated classification technique (Mingo et al. in preparation) involved using masked 4rms arrays (where emission below 4rms is removed and potential unassociated emission is masked), rather than the raw FITS data. We define unassociated emission as radio emission which does not appear to belong to the radio source in question. A flood-filling algorithm[2] and masking techniques have additionally been applied in order to identify and use associated structures and consequently remove unassociated emission from the image (Mingo et al. in preparation). On the other hand, this work emphasizes using the raw FITS images as the input to the deep

[2]http://scikit-image.org/docs/dev/api/skimage.measure.html#skimage.measure.label

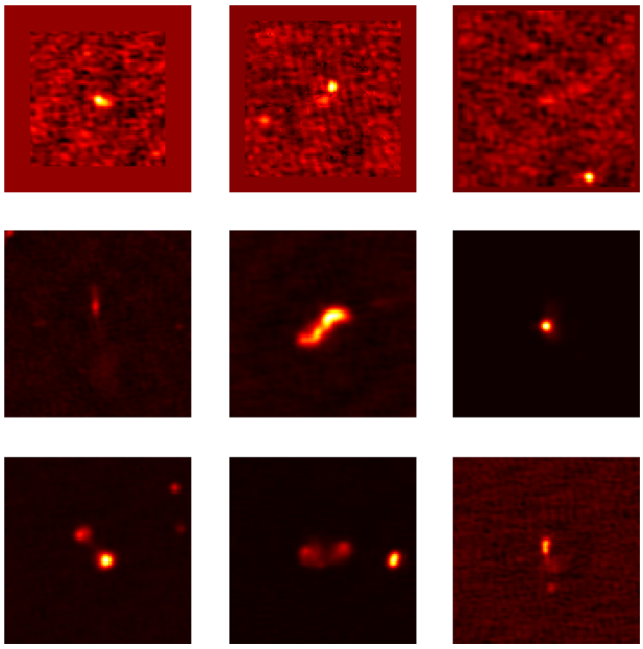

Figure 3. Showing morphology samples of the FITS cut-outs when converted to png images using the 'hot' colourmap. The top row shows the 'Unresolved class', middle row shows the FRI class, and the bottom row shows FRII. There are varying levels of noise and the occasional potentially unassociated emission present in the images.

learning algorithms, to see if they could be trained to cope with unassociated emission and unfiltered noise. After visual inspection we found there were approximately 1 per cent of images containing potentially unassociated emission, whereas the majority of the images contain varying levels of noise.

In cases where the calibration did not perform as expected, the source will not be de-convolved accurately, causing flux leakage. This could result in the source being misclassified, leading to label errors. After inspecting several batches of images, we estimated the amount of labels containing errors to be less than 6 per cent, when considering both FRIs and FRIIs. Since larger sources are easier to classify, there is a decreased likelihood that they will be mislabelled, therefore the size of the source affects the presence of noisy labels. However, pre-filtering is applied to ensure the effect is not very large.

Fig. 3 shows typical examples of source types across the three classes. It is evident that there are varying levels of noise present in the images, presenting the largest hindrance to the deep learning algorithms' ability to classify the sources accurately. One of the aims of this work is to see how well the algorithms can classify the sources in the presence of such undesirable features, present in the original radio images (FITS files). We also compare the results obtained when using the masked 4rms clipped arrays (see Section 5.3), where emission below 4rms is removed and potential unassociated emission is masked.

3 METHODS

We use the radio galaxy image FITS cut-outs from version 1.0 of the LoTSS DR1 value-added catalogue (Williams et al. 2019). The extended source identifications do not differ from the final version to a large extent.

3.1 Pre-processing

Since the size of each cut-out varies, they first need to be made the same size. The FITS images have been resized to (200,200) pixels, where the smaller images have been padded with zeros around the edges, and the larger images have been downsampled, using bicubic interpolation. The sizes of the arrays vary across all three classes. Following this, the images are centred on the position of the optical source, ensuring its position is at (100,100). We crop to the inner (100,100) pixel part of the image as the source is likely to be contained in this interval and to reduce the amount of data input into the network. The pixel values, representing brightness in mJy beam^{-1}, were normalized by dividing by the maximum value in each image, therefore the values are contained within the [0,1] range. The images are taken at 150 MHz. We apply the 'hot' colourmap from the PYTHON MATPLOTLIB library, which converts the images from a single channel numpy array to an RGB png image. This is done by assigning a colour (RGB vector) according to the value in the single channel array. For example, values close to 1 are bright yellow in the 'hot' colourmap scheme, therefore (r,g,b) $\approx$ (1,1,0.99). The conversions to the RGB vector are provided.[3] The conversion is done to make the arrays more amenable to deep learning analysis and has no bearing on the flux values. The number of sources in each class is given in Table 1.

Cropping the images to (100,100) pixels, instead of using the originally resized images of (200,200) pixels, reduces the impact of radio emission that is potentially unassociated with the main source in the centre. We have also experimented with using central sizes other than (100,100) pixels, however they resulted in worsened performance metrics. Smaller images tended to have some associated emission truncated, whereas larger images encapsulated more unassociated emission. The cropping still preserved the general noise characteristics surrounding the source.

The upsizing of images should not have any detrimental effects on image quality, however the downsizing may cause effects such as slight distortion of the radio emission due to the interpolation.

3.2 Image augmentation

Deep learning algorithms generally require large numbers of labelled images in order to make predictions more successfully and to reduce the effect of overfitting, in which the algorithm memorizes the training samples and therefore the model fails to generalize on an independent data set. More images can be generated artificially, by performing simple transformations to the original data (Krizhevsky, Sutskever & Hinton 2012). As such, we apply translation, rotation, and flipping to generate more images. In using translation, we initially use a random number that shifts the image between 0 and 20 pixels in any of the four directions, using the condition that if such a translation moves the brightest pixel out of the image, the translation is reduced to 10 per cent of the original value. This is to reduce the possibility that part of a radio component will be shifted out of the image. The images have been rotated randomly in multiples of 90 deg only in order to avoid interpolation artefacts. We note that since there is a limited range of rotation applied, it is not enough to ensure complete rotational invariance in our models. Both horizontal and vertical flipping have been applied at random. The augmentation of the FRI and FRII sources has been done keeping their overall

[3] y = (0,0.36): (r,g,b) $\approx$ (x = y/0.36,0,0).y = (0.36,0.74): (r,g,b) $\approx$ (1,x = (y−0.37)/0.37,0).y = (0.74,1): (r,g,b) $\approx$ (1,1,x = (y−0.75)/0.25).

proportions similar in number to the original data set as this resulted in improved performance. The number of original and augmented images used in this work is given in Table 1. Image augmentation is applied on both the original LOFAR images, as well as the masked 4rms arrays.

4 DEEP LEARNING ALGORITHMS

The most successful class of machine learning methods in the context of extracting information from high-dimensional data is deep learning, which has achieved unprecedented performance in a variety of domains such as image recognition, sentiment analysis, and genomics (LeCun, Bengio & Hinton 2015). Their ability to learn multiple representations of data lies in their stacked layer architecture. The most commonly used implementation of deep learning has to date been convolutional neural networks. However, more recent advances were made in addressing the lack of rotational invariance in convolutional neural networks through the development of capsule networks.

4.1 Convolutional neural networks

Neural networks and deep learning algorithms are generally trained using the backpropagation algorithm, where a gradient descent optimization algorithm is used to minimize the error between the predictions of the network and the input labels by calculating the gradients and adjusting the weights accordingly (Rumelhart, Hinton & Williams 1986). A deep fully connected neural network becomes time consuming and computationally intensive to train. Convolutional neural networks employ smaller sized filters that scan across the image and extract features, which greatly reduces the dimensionality compared to using adjacent layers of fully connected neurons and enforces parameter sharing and therefore translational invariance (Karpathy 2016). Spatial pooling layers are typically inserted between at least one convolutional layer which further reduces the dimensionality of features propagated through the network. In max pooling, the maximum value of a certain region of the image is output into the next layer. However, since the pooling operation summarizes the information in a local part of the image, the global feature information within the image tends to degrade.

4.2 Capsule networks

Capsule networks (Sabour, Frosst & Hinton 2017) have been developed to preserve the relative locations of features within images and thus model the hierarchical relationships better. Although traditional neural networks output a single activation value, capsule networks are higher dimensional and output a vector representing a group of parameters such as orientation, skew, thickness etc., depending on the input. The overall length of these vectors give the probability that the entity exists. Capsule networks have achieved state-of-the-art performance on the MNIST data set (LeCun et al. 1998) without data augmentation (Xi, Bing & Jin 2017).

In the context of radio galaxy classification, capsule networks should be able to preserve the emission pattern features over a large spatial extent, given an adequate training set size.

Below we summarize the theory behind capsule networks but see Sabour et al. (2017) for a detailed description. For all capsules above the first layer of capsules, the input to a capsule s_j is a weighted sum over all prediction vectors from the capsules in the layer below, given by multiplying the coupling coefficients c_{ij} by the output u_i of a capsule in the layer below by a weight matrix W_{ij}, as shown in equation (1)

$$s_j = \sum_i c_{ij} W_{ij} u_i. \tag{1}$$

The coupling coefficients c_{ij} are determined by a routing softmax function given by equation (2)

$$c_{ij} = \frac{e^{b_{ij}}}{\sum_k e^{b_{ik}}}. \tag{2}$$

The coupling coefficient c_{ij} is the level of agreement between the predicted output of capsules in a layer, to their parent capsules in the layer above. b_{ij} gives the log prior probabilities that capsule i should be coupled to capsule j.

The vector length is calculated as shown in equation (3)

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||}, \tag{3}$$

where v_j is the vector output of capsule j and s_j is its total input. This output gives the probability that a specific property exists in the input to the capsule that is represented by the capsule. The vector output v_j is an activation function that is also referred to as a squashing function as it shrinks short vectors to near zero if a property is not present in the capsule, and long vectors to lengths close to 1 if the property exists.

The agreement a_{ij} for updating log probabilities and coupling coefficients is given by equation (4)

$$a_{ij} = v_j . W_{ij} u_i. \tag{4}$$

A margin loss function is used in order to determine whether a radio galaxy of a particular class is present, which has the form given by equation (5):

$$L_k = T_k \, max(0, m^+ - ||v_k||)^2 + \lambda(1 - T_k) \, max(0, ||v_k|| - m^-)^2, \tag{5}$$

where $T_k = 1$ if a radio galaxy of class k is present and $m^+ = 0.9$ and $m^- = 0.1$, to ensure that the vector length remains within reasonable bounds. The λ down-weighting function is introduced for numerical stability and suggested to be set at 0.5.

The mean squared error difference between the reconstructed image from the decoder (the part of the Capsule network after LabelCaps) and the input image acts as a regularizer for the capsule network, such that near-perfect reconstructions will produce a near-zero error and poor reconstructions will produce a large error. The reconstruction loss is scaled down by 0.0005 so it does not dominate the margin loss during training, and the coefficient for the default model is designed for the MNIST digits which have an image size of 28×28, thus the coefficient is worked out to be $0.0005 \times 28 \times 28 = 0.392$.

The architecture and number of parameters in the default Capsule network used in the current work is shown in Table 2.

4.3 Deep learning parameters

There are several deep learning implementations currently available for use. This work uses Keras[4] with the TensorFlow[5] backend and PYTHON version 2.7.14.

[4] https://keras.io/preprocessing/image/.
[5] https://www.tensorflow.org.

Table 2. Showing architecture for the default capsule network model.

Layer	Output shape	# Params
Input_1	(None, 100, 100, 3)	0
conv2d	(None, 92, 92, 256)	62 464
PrimaryCap_conv2d	(None, 42, 42, 6)	124 422
PrimaryCap_reshape	(None, 3528, 3)	–
PrimaryCap_squash	(None, 3528, 3)	–
LabelCaps	(None, 3, 3)	95 256
Input_2	(None, 3)	–
mask	(None, 9)	–
capsnet	(None, 3)	–
decoder	(None, 100, 100, 3)	3878 960
Total		4161 102

Table 3. ConvNet-4 architecture. A filter size of 5 is used in the convolutional layers.

Layer	Output shape	# Params
Input	(None, 100, 100, 3)	0
conv2d	(None, 100, 100, 16)	1216
conv2d	(None, 100, 100, 16)	6416
maxpool2d	(None, 50, 50, 16)	–
dropout	(None, 50, 50, 16)	–
conv2d	(None, 50, 50, 16)	6416
conv2d	(None, 50, 50, 16)	6416
maxpool2d	(None, 25, 25, 16)	–
dropout	(None, 25, 25, 16)	–
flatten	(None, 10000)	–
dense	(None, 500)	5000 500
dropout	(None, 500)	–
dense	(None, 3)	1503
Total		5022 467

We use the Adam optimizer (Kingma & Ba 2014) with the default learning rate of 0.001. In order to keep more parameters the same between the models, both the convolutional and capsule network models are trained using a batch size of 100, for 50 epochs.

The deep learning task is a multiclassification problem, where the models output a three-dimensional vector representing the probability that the object belongs to each class. The predicted class is chosen as the one with the largest probability value. As the probabilities are independent, there is no constraint that they need to add to unity.

The models are trained using CPUs from 27 available Intel XEON CPU nodes with six available cores per node on a computing cluster at the University of Hamburg.

4.3.1 *ConvNet-4 parameters*

We use an architecture of two pairs of stacked convolutional layers with pooling layers in between, as shown in Fig. 4, with parameters given in Table 3. This model is referred to as ConvNet-4. Using two adjacent convolutional layers with smaller filter sizes obtained improved results compared to using a single larger convolutional layer, and also reduced the number of parameters (Simonyan & Zisserman 2015). We use the categorical cross-entropy cost function[6] and 16 filters of size 5 × 5 across all layers, as well as the default learning rate decay of 0. In order to reduce the effect

[6]https://keras.io/losses/#categorical_crossentropy.

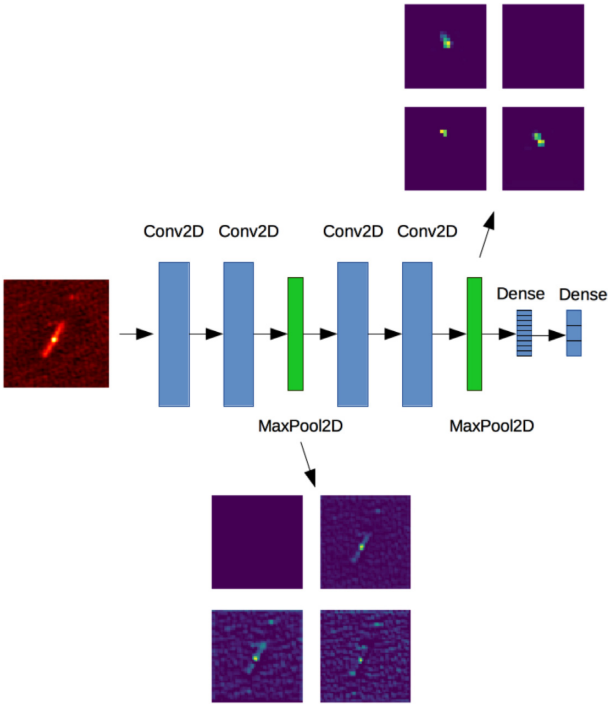

Figure 4. The ConvNet-4 architecture. The input to the network is a 100 × 100 × 3 image. Showing an example input image with features detected at the second and fourth convolutional layers, after pooling, at the end of training (50 epochs). We show four feature maps for each of the two outputs.

of overfitting, dropout layers are used. A dropout value of 0.25 is used after each pair of convolutional layers, and a value of 0.5 in between the dense layers. A penalty term is added to the cost function using L2 regularization (Ng 2004) in the first dense layer. All the convolutional layers use the ReLU activation function (Nair & Hinton 2010), and the softmax activation function at the final layer where classifications are made. There are 5022 467 trainable parameters in total.

4.3.2 *ConvNet-8 parameters*

In order to investigate the performance for more complex convolutional networks, we can add additional layers. The ConvNet-8 model uses an architecture of four pairs of stacked convolutional layers with pooling layers in between. There are also an increasing number of feature maps with each subsequent double stacking of convolutional layers, as shown in Table 4. The architecture also uses smaller feature maps of size 3 × 3. There are 7446 259 trainable parameters in total.

4.3.3 *CapsNet parameters*

Finally, we explore several variations of capsule network models. We downloaded the original CAPSULENET[7] code implemented in Keras that was built for the MNIST data set (Sabour et al. 2017), and modified the code to use our data sets, vary the models from the original architecture, and to calculate the metrics. The original architecture contains approximately 58M parameters, which is more

[7]https://github.com/XifengGuo/CapsNet-Keras/blob/master/capsulenet.py.

Table 4. ConvNet-8 architecture. A filter size of 3 is used in the convolutional layers.

Layer	Output shape	# Params
Input	(None, 100, 100, 3)	0
conv2d	(None, 100, 100, 32)	896
conv2d	(None, 100, 100, 32)	9248
maxpool2d	(None, 50, 50, 32)	–
dropout	(None, 50, 50, 32)	–
conv2d	(None, 50, 50, 64)	18 496
conv2d	(None, 50, 50, 64)	36 928
maxpool2d	(None, 25, 25, 64)	–
dropout	(None, 25, 25, 64)	–
conv2d	(None, 25, 25, 128)	73 856
conv2d	(None, 25, 25, 128)	147 584
maxpool2d	(None, 13, 13, 128)	–
dropout	(None, 13, 13, 128)	–
conv2d	(None, 13, 13, 256)	295 168
conv2d	(None, 13, 13, 256)	590 080
maxpool2d	(None, 7, 7, 256)	–
dropout	(None, 7, 7, 256)	–
flatten	(None, 12544)	–
dense	(None, 500)	6272 500
dropout	(None, 500)	–
dense	(None, 3)	1503
Total		7446 259

than 14× the number of parameters as for the ConvNet-4 model. We therefore simplified the architecture to one having just over 4M parameters, and refer to this as the default model. The original CapsuleNet model is simplified in order to have the same order of magnitude as the parameters in the ConvNets and to help prevent overfitting.

The default architecture of CapsNet and decoder is illustrated in Fig. 5 and the number of parameters is given in Table 2. In essence it is comprised of an encoder and decoder. The encoder consists of a convolutional layer, which extracts features in the image, which are then input into the first capsule layer (PrimaryCaps), whose function is to take the 256 × 9 × 9 output of the convolutional layer and produce combinations of the detected features. The output of the PrimaryCaps layer is then sent to the LabelCaps layer, which produces one three-dimensional capsule for each of the three radio galaxy classes. Routing is used between the PrimaryCaps layer and the LabelCaps layer such that the level of agreement of feature existence can be quantified and contribute to the vector length of the capsule. The decoder refers to the part of the network after the LabelCaps layer (the three dense layers at the end). There are 4161 102 free parameters in the default CapsNet model.

We use 256 filters in the first convolutional layer, a filter size of 9 in both the first Convolutional layer and PrimaryCaps layer, three capsules in the PrimaryCaps and LabelCaps layers, two channels in the PrimaryCaps and the decoder contains (64 ,128) nodes. We use the default set-up of three routings and a learning rate of 0.001 with a decay of 0.9. The first convolutional layer uses the ReLU activation function. CapsNet has image augmentation built into the training of the model, which we disable in order to use our augmentation technique, that allows more control over which classes get augmented and the type of transformations that are used. For the default CapsNet model, there are 4161 102 parameters, which is a very similar number of parameters that was used for ConvNet-4.

In addition to the default CapsNet model, we experiment with two other CapsNet models. In the first of these models (Inc. filtersize), we set the filter size to 24 and 18 in the first Convolutional layer and PrimaryCaps layer, respectively, and slide the filters across using a stride of 4 in the convolutional layer. The inc. filtersize model has 4819 470 parameters. In the second model (Inc. decoder), we increase the complexity of the decoder to (128, 256) nodes in the dense layers and the loss function of the decoder weight is increased from 0.392 to 5, respectively. The weight is calculated by taking the scaled-down reconstruction loss and multiplying it by the size of the

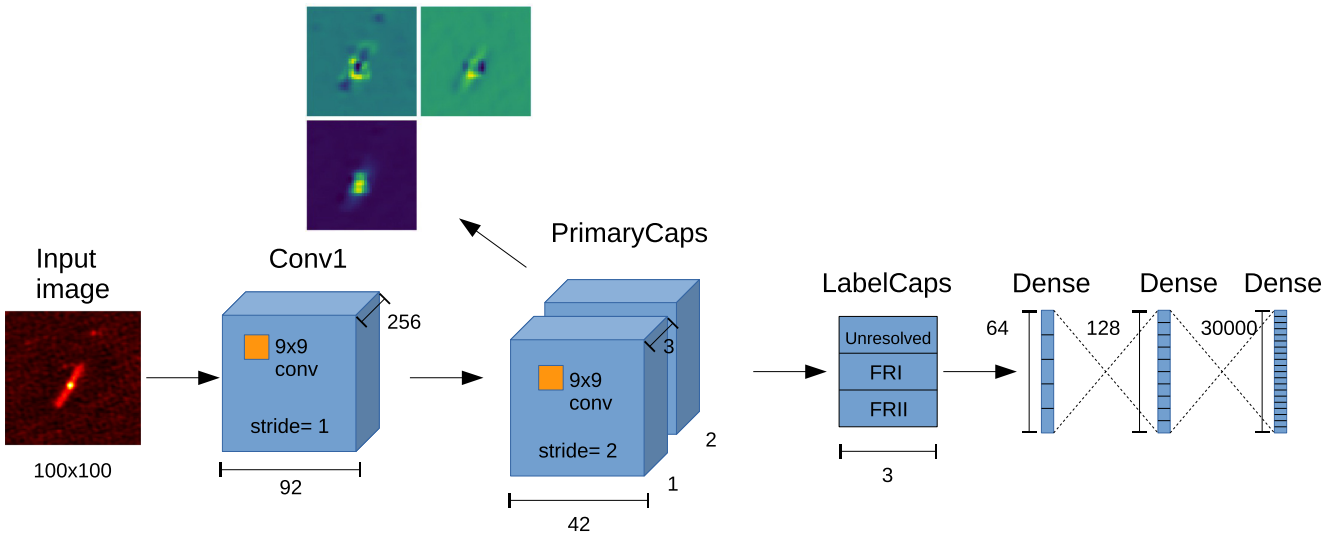

Figure 5. The default architecture for CapsNet, using three classes. The input to the network is a 100 × 100 × 3 image. The encoder is the part of the network that encapsulates the convolutional layer up to and including the LabelCaps layer. The decoder refers to the final three dense layers. An example of features detected by the PrimaryCaps layer prior to reshaping and squashing is shown, for the given input image. There is a small amount of extended emission to the top right of the image that appears to be unassociated with the main source in the centre, which the capsule network preserves, suggesting that it is not robust to potential unassociated sources. Additionally, the feature maps appear to show extra distortion in the core of the source.

images $0.0005 \times 100 \times 100 = 5$. There are 8026 446 parameters in the inc. decoder model.

We chose to increase the filters from a size of 9 pixels in the inc. filtersize model because the original filter sizes that were designed for the MNIST image sizes of (28,28) pixels are likely too small compared to what would be needed for our (100,100) pixel images. We also experimented with increasing the number of nodes and weight loss of the decoder in the inc. decoder model to better account for the noise and potential unassociated emission in the data set, as well as more variability in and between classes.

5 RESULTS

Due to the inherent stochasticity of training deep learning models, each run can produce slightly different results. We therefore train each model five times. The training data are also shuffled for each run to ensure there is no correlation between subsequent samples. There are several classification metrics that can help evaluate the performance of a classifier. In imbalanced class problems, the classification accuracy alone has several weaknesses in distinguishing between the performance of models (Hossin & Sulaiman 2015). The precision, recall, and F1 scores are more informative measures of performance compared to using the classification accuracy. Precision refers to the fraction of true positives returned among all returned positive instances, recall is the fraction of true positives that are identified correctly, which also gives an indication of the sensitivity of the classifier. The F1 score is the harmonic mean of precision and recall, and can be interpreted as the average of the precision and recall values. The accuracy is the total proportion of correct predictions. Precision, recall, F1 score, and accuracy are defined in equations (6)–(9).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1\_score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{9}$$

where TP refers to the true positives, FP refers to the false positives, and FN refers to false negatives. A true positive is when the prediction matches the label. A false positive is when the positive class is incorrectly predicted. A false negative is when the positive class is predicted to be in another class.

We also calculate the 95 per cent confidence interval using the mean and standard deviation of the metrics to account for the variability in performance across the runs. We declare a model to be statistically significantly better than another model if the mean of its metrics is higher than the 95 per cent confidence interval of the other models metrics. In order to ensure a fair comparison, the same training and testing sets were used for the ConvNet and CapsNet architectures.

The same set of data is used for both validation and testing when running the models, with the exception of the application of early stopping (results shown in Section 5.4.4). When early stopping is used, the validation data are used to determine when to stop the training. Otherwise, the use of the same data set for validation and testing is of no consequence, as the weights that are modified using the training set are applied to the validation/test set to calculate the loss. No adjustment is made to the weights using the validation set. At the conclusion of training, the final weights are applied to the validation/test set and the metrics are calculated.

Table 5. The average metrics (in percentages) across each of the classes in (1) the original LOFAR data set, (2) the original and augmented data set, (3) the original 4rms clipped data set, and (4) the original and augmented 4rms clipped data set for the ConvNet-4 model. Five runs were done in total, using 600 samples in the test set.

Class	Precision	Recall	F1 score	Accuracy
(1)				
Unres.	95.7 ± 0.9	96.7 ± 1.4	96.2 ± 0.9	95.9 ± 0.9
FRI	86.2 ± 2.4	86.8 ± 1.1	86.5 ± 1.0	89.9 ± 0.9
FRII	68.0 ± 1.1	63.5 ± 2.1	65.6 ± 1.0	90.9 ± 0.2
Avg.	88.5 ± 0.8	88.7 ± 0.8	88.6 ± 0.9	93.1 ± 0.8
(2)				
Unres.	98.1 ± 0.4	98.2 ± 0.5	98.1 ± 0.4	98.0 ± 0.4
FRI	92.3 ± 0.9	93.3 ± 1.3	92.3 ± 0.2	94.2 ± 0.1
FRII	80.9 ± 2.0	75.2 ± 4.9	77.8 ± 1.9	94.2 ± 0.2
Avg.	93.3 ± 0.2	93.4 ± 0.2	93.3 ± 0.2	96.2 ± 0.2
(3)				
Unres.	97.9 ± 0.3	98.1 ± 0.5	98.0 ± 0.2	97.9 ± 0.2
FRI	90.4 ± 0.7	90.0 ± 0.6	90.2 ± 0.4	92.8 ± 0.3
FRII	72.1 ± 0.6	72.2 ± 1.6	72.1 ± 0.8	92.5 ± 0.2
Avg.	91.8 ± 0.2	91.9 ± 0.3	91.8 ± 0.3	95.5 ± 0.2
(4)				
Unres.	98.7 ± 0.6	99.7 ± 0.2	99.2 ± 0.2	99.2 ± 0.2
FRI	91.5 ± 0.9	94.9 ± 0.6	93.1 ± 0.4	94.9 ± 0.3
FRII	88.1 ± 1.3	75.5 ± 2.3	81.3 ± 1.4	95.3 ± 0.3
Avg.	94.9 ± 0.2	94.7 ± 0.3	94.7 ± 0.2	97.3 ± 0.1

Section 5.1 of the results shows the classification metrics across the two deep learning techniques when using the original data only, with 2301 (79 per cent) samples for training, and 600 (21 per cent) samples for both validation and testing. The fraction of samples in each class is given in Table 1 for the test set. Section 5.2 uses augmented images in addition to the original images and Section 5.3 explores the effects when the 4rms sigma-clipped data are used.

5.1 LOFAR original images

5.1.1 ConvNet-4 and ConvNet-8 models

We use the ConvNet-4 and ConvNet-8 models on the original 2901 images from LOFAR, which have been classified into Unresolved, FRI, and FRII sources. The results are shown in Tables 5 and 6. Each epoch consisting of 2301 training samples takes approximately 32 and 66 s to train for ConvNet-4 and ConvNet-8, respectively.

The models perform the best in recovering the images in the Unresolved class, which could be due to the images being generally noisier and the sources smaller, compared to the other images. The recovery of FRIIs is poorer however compared to the FRIs. This may be because there are fewer examples of images in this class (460 FRIIs compared to 984 FRIs). Although it can be argued that the morphological diversity is greater for the FRI class as they can be straight, bent, or one-sided with a peak at one end, FRIIs contain lobes that may or may not be connected, therefore the source can contain either one or two components. We have experimented with

Table 6. The average metrics (in percentages) across each of the classes in (1) the original LOFAR data set, (2) the original and augmented data set, and (3) the original 4rms clipped data set, and (4) the original and augmented 4rms clipped data set, for the ConvNet-8 model. Five runs were done in total, using 600 samples in the test set.

Class	Precision	Recall	F1 score	Accuracy
(1)				
Unres.	96.6 ± 1.1	98.8 ± 0.3	97.7 ± 0.5	97.5 ± 0.6
FRI	88.7 ± 1.0	90.4 ± 0.7	89.6 ± 0.5	92.2 ± 0.4
FRII	75.2 ± 4.1	64.5 ± 2.8	69.3 ± 1.1	92.3 ± 0.4
Avg.	90.9 ± 0.4	91.2 ± 0.4	90.9 ± 0.5	94.9 ± 0.4
(2)				
Unres.	98.2 ± 0.7	98.4 ± 0.2	98.3 ± 0.3	98.2 ± 0.3
FRI	92.5 ± 0.6	94.0 ± 0.5	93.2 ± 0.4	95.0 ± 0.3
FRII	84.5 ± 1.9	80.0 ± 1.0	82.2 ± 1.1	95.3 ± 0.3
Avg.	94.3 ± 0.2	94.3 ± 0.2	94.3 ± 0.2	96.7 ± 0.1
(3)				
Unres.	99.6 ± 0.3	98.8 ± 1.0	99.2 ± 0.5	99.1 ± 0.5
FRI	92.7 ± 1.0	93.4 ± 3.3	93.0 ± 2.1	95.2 ± 1.7
FRII	83.4 ± 9.3	83.4 ± 2.8	83.1 ± 5.8	95.2 ± 1.8
Avg.	95.0 ± 1.6	94.9 ± 1.8	94.9 ± 1.7	97.3 ± 1.0
(4)				
Unres.	99.6 ± 0.1	99.1 ± 0.4	99.3 ± 0.2	99.3 ± 0.2
FRI	94.4 ± 0.4	95.2 ± 0.7	94.8 ± 0.4	96.2 ± 0.3
FRII	86.0 ± 1.0	85.8 ± 1.5	85.9 ± 0.6	96.2 ± 0.1
Avg.	96.0 ± 0.2	95.9 ± 0.2	95.9 ± 0.2	97.9 ± 0.2

using different weights for the classes, giving proportionally greater weights for the FRIIs such that wrong predictions are penalized more, however the performance remained the same as before, across all classes. The recall tends to be higher compared to precision for the FRIs, whereas it is lower compared to precision for the FRIIs. This is likely due to it being easier to recover sources containing emission that is more concentrated in one place (in the case of the FRIs), compared to emission that is further apart.

Examples of detected features in the ConvNet-4 model at the output of the second and fourth convolutional layers, after max pooling are shown in Fig. 4. The training and validation losses for a single run with the ConvNet-4 architecture are shown in Fig. 6.

The use of a more complex architecture (ConvNet-8 compared to ConvNet-4) appears to improve the classification metrics (Avg. Recall = 91.2 compared to 88.7, respectively).

5.1.2 *CapsNet model*

Each epoch consisting of 2301 training samples takes approximately 3.4 min for the default model, 14 s for the inc. filtersize model and 3.5 min for the inc. decoder model. The faster time for the inc. filtersize model is due to the fact that the feature maps are moved across the image by 4 pixels (stride of 4) in the first convolutional layer as opposed to using a stride of 1, therefore the feature maps are able to scan through the image faster.

Examples of detected features at the PrimaryCaps layer, prior to the reshape and squashing functions are shown in Fig. 5 for the default model. Fig. 7 shows the training and validation loss curve for the default model. Table 7 shows that the default model attains higher overall metrics compared to the other two CapsNet models (although this is not always significant).

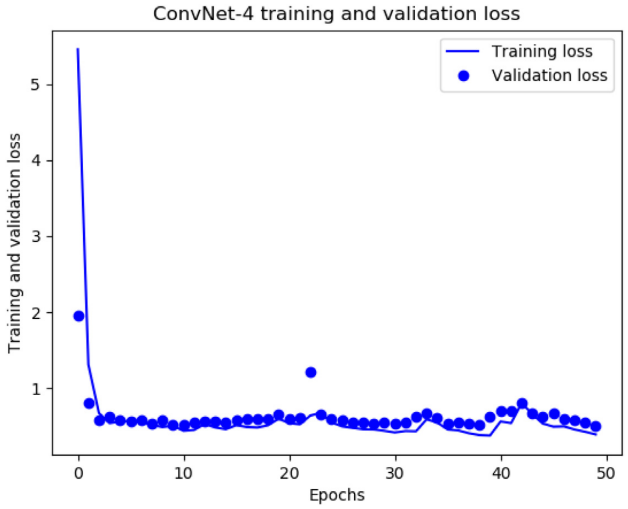

Figure 6. The training and validation losses for a single run with the ConvNet-4 architecture using the cross-entropy loss, with 2301 (79 per cent) samples for training and 600 (21 per cent) samples for testing.

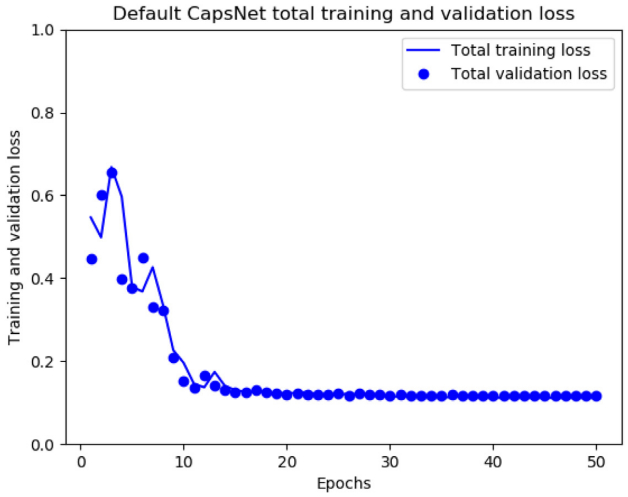

Figure 7. The training and validation losses for a single run with the default capsule network architecture, using the margin loss as defined in equation (5), with 2301 (79 per cent) samples for training and 600 (21 per cent) samples for testing. The total loss is obtained by adding the capsule network loss to the decoder weight multiplied by the decoder loss.

The inc. filtersize model, which was designed with larger filters to capture more extended emission, for the most part performs as well as the default model and the metrics for the FRIIs are improved. However, they tend to be lower for the Unresolved and FRI classes, which make up the majority of samples. The results are shown in Table 8.

The inc. decoder model, which uses a more complex decoder, performs as well as the default model in the metrics for the Unresolved and FRI classes. However, it performs worse overall for the FRIIs, as shown in Table 9. This may be due to the more complex decoder confusing radio emission from the FRIIs with noise.

As the default CapsNet model performs better overall compared to the other two CapsNet models, it is chosen as the basis of comparison against the two ConvNet models across the original FITS and masked 4rms sigma-clipped data sets.

Table 7. The average metrics (in percentages) across each of the classes in the original LOFAR data set, for the default CapsNet model. Five runs were done in total, using 600 samples in the test set.

Class	Precision	Recall	F1 score	Accuracy
(1)				
Unres.	92.7 ± 1.4	95.7 ± 0.7	94.2 ± 1.0	93.4 ± 1.2
FRI	78.3 ± 3.1	87.7 ± 1.6	82.7 ± 1.1	86.3 ± 1.3
FRII	66.6 ± 5.1	35.0 ± 13.0	43.1 ± 12.7	88.2 ± 0.8
Avg.	84.0 ± 1.3	84.7 ± 1.5	83.2 ± 2.6	90.1 ± 1.2
(2)				
Unres.	96.4 ± 0.6	96.4 ± 0.9	96.4 ± 0.2	96.1 ± 0.2
FRI	85.5 ± 1.4	90.2 ± 0.2	87.8 ± 0.7	90.7 ± 0.6
FRII	75.8 ± 1.8	64.2 ± 0.5	69.6 ± 1.4	92.3 ± 0.4
Avg.	89.7 ± 0.5	89.9 ± 0.5	89.7 ± 0.5	93.7 ± 0.3
(3)				
Unres.	97.3 ± 0.5	98.1 ± 0.1	97.7 ± 0.3	97.5 ± 0.3
FRI	90.9 ± 0.7	88.4 ± 0.8	89.6 ± 0.6	92.5 ± 0.5
FRII	72.0 ± 2.6	75.2 ± 3.3	73.6 ± 2.8	92.7 ± 0.8
Avg.	91.6 ± 0.7	91.5 ± 0.7	91.5 ± 0.7	95.0 ± 0.4
(4)				
Unres.	98.4 ± 0.1	98.3 ± 0.1	98.3 ± 0.0	98.3 ± 0.0
FRI	92.0 ± 0.6	91.3 ± 1.2	91.7 ± 0.5	93.9 ± 0.4
FRII	80.4 ± 2.4	82.3 ± 1.8	81.2 ± 1.1	94.9 ± 0.4
Avg.	93.7 ± 0.3	93.6 ± 0.4	93.6 ± 0.3	96.2 ± 0.2

Table 8. The average metrics (in percentages) across each of the classes in the original LOFAR data set, for the inc. filtersize CapsNet model. Five runs were done in total, using 600 samples in the test set.

Class	Precision	Recall	F1 score	Accuracy
Orig.				
Unres.	89.6 ± 0.7	94.2 ± 0.3	91.8 ± 0.5	90.8 ± 0.5
FRI	80.4 ± 2.5	79.6 ± 2.9	79.9 ± 0.1	85.0 ± 0.5
FRII	63.2 ± 6.4	50.5 ± 10.8	54.2 ± 6.7	88.4 ± 0.2
Avg.	82.7 ± 0.5	83.0 ± 0.5	82.5 ± 1.1	88.4 ± 0.4

Table 9. The average metrics (in percentages) across each of the classes in the original LOFAR data set, for the inc. decoder CapsNet model. Five runs were done in total, using 600 samples in the test set.

Class	Precision	Recall	F1 score	Accuracy
Orig.				
Unres.	90.6 ± 2.7	95.0 ± 0.8	92.7 ± 1.8	91.6 ± 2.2
FRI	75.1 ± 2.5	87.8 ± 1.9	80.9 ± 2.2	84.5 ± 1.9
FRII	65.8 ± 2.9	22.7 ± 9.0	32.3 ± 10.7	87.4 ± 0.8
Avg.	81.6 ± 2.0	82.7 ± 2.1	80.3 ± 3.0	88.5 ± 1.9

The default CapsNet model still performs significantly worse compared to the two ConvNets, as it is beyond both their 95 per cent confidence intervals, across all metrics. The variability in metrics is higher for the original data set compared to that of the two ConvNets, as is evident in the generally increased confidence intervals of the CapsNet model, in Table 7, particularly for the FRIIs.

Fig. 8 shows the Receiver Operating Characteristic (ROC) curves across the default capsule network and ConvNet-4. ROC curves plot the true positive rate (recall) against the false positive rate.

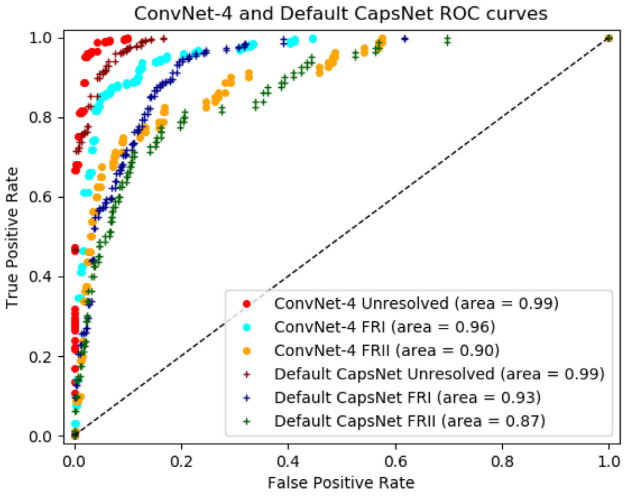

Figure 8. ROC curves for both a single run with the default CapsNet model and the ConvNet-4 model. The curves show that ConvNet-4 outperforms the default CapsNet across all the classes.

Table 10. The labels and corresponding probability vector of the default CapsNet network predictions, using four examples of sources shown in Fig. 10, having probabilities greater than 0.5 across two classes.

Source	Label	Probability vector (Unres., FRI, FRII)
1	FRI	(0.41, 0.5 , 0.51)
2	Unres.	(0.51, 0.36 , 0.62)
3	FRII	(0.34 , 0.59 , 0.57)
4	FRII	(0.16 , 0.72 , 0.7)

In a first attempt to use the default CapsNet model (containing 58M free parameters), we observed a clear overfitting, owing to the large number of free parameters compared to the number of training images. Despite this, the model still achieved very similar results to the models using many fewer parameters quoted in this work.

Table 10 shows the labels and prediction vector for some sources that the Capsule network could not reliably classify , as probabilities higher than 0.5 across two classes were attained.

5.2 LOFAR original and augmented images

We augmented the images with translation, rotation, and flipping as outlined in Section 3.2, keeping the distribution of FRI and FRII sources the same as in the original data set. Table 1 gives the number of original and augmented images. There are again 79 per cent and 21 per cent of the original samples used in training and testing, respectively.

5.2.1 ConvNet-4 and ConvNet-8

We applied both ConvNet-4 and ConvNet-8 models to the original and augmented data set, with the results shown in Tables 5 and 6. The overall metrics are significantly better (Avg. Recall = 93.4 and 94.3) than was observed when the same model was used on the original images (Avg. Recall = 88.7 and 91.2 for ConvNet-4 and ConvNet-8, respectively), therefore both models benefit from data augmentation. The confidence intervals are also usually reduced.

Table 11. Confusion matrix for a single run with the ConvNet-4 architecture, after training on the original and augmented images. The predictions are along the columns and the labels are along the rows.

	Unres.	FRI	FRII	Total
Unres.	294	6	1	301
FRI	3	202	14	219
FRII	5	12	63	80
Total	302	220	78	600

Table 12. Confusion matrix for a single run with the default CapsNet architecture, after training on the original and augmented images. The predictions are along the columns and the labels are along the rows.

	Unres.	FRI	FRII	Total
Unres.	289	12	0	301
FRI	4	198	17	219
FRII	6	24	50	80
Total	299	234	67	600

Although the classification metrics remain the poorest for the FRII class, they improved the most when using the augmented data, despite the fact that there were more examples of FRIs.

A confusion matrix is provided in Table 11 for the ConvNet-4 model, to see the numbers of samples that are both correctly and incorrectly predicted.

5.2.2 *CapsNet*

The best-performing capsule network (the default model) was used to see whether an improvement in overall metrics could be obtained when using augmented images in addition to the original images. The results are shown in Table 7. The confusion matrix for a single run with the default CapsNet architecture, after training on the original and augmented images, is given in Table 12.

The classification metrics are significantly improved when using the augmented images (Avg. Recall = 89.3 with augmentation, compared to Avg. Recall = 84.2 ± 0.2 without), therefore the capsule network also benefits from training on additional images. Despite the fact that capsule networks output a vector describing the properties of images across the classes and aim to extract the underlying patterns, they still benefit from the use of additional augmented images, for the FITS file data set. The noise in the images could be preventing the network from seeing the underlying morphology in the signal, and there is an insufficient number of images available across the classes, hence improved results are observed when more examples are provided. Despite CapsNet benefitting from augmentation, the classification metrics are still significantly lower compared to when augmentation is applied to the two ConvNets.

Fig. 9 shows the real and reconstructed images for a single run of the default CapsNet model when training on the original and augmented images. The labels match the predictions with the exception of the third and fourth images in the top two rows, where the true labels are FRIIs but the predictions are FRIs. The reconstructions of the images are innaccurate, giving the appearance that CapsNet is determining class membership based on the blurriness of the reconstructed spheres. The images in the

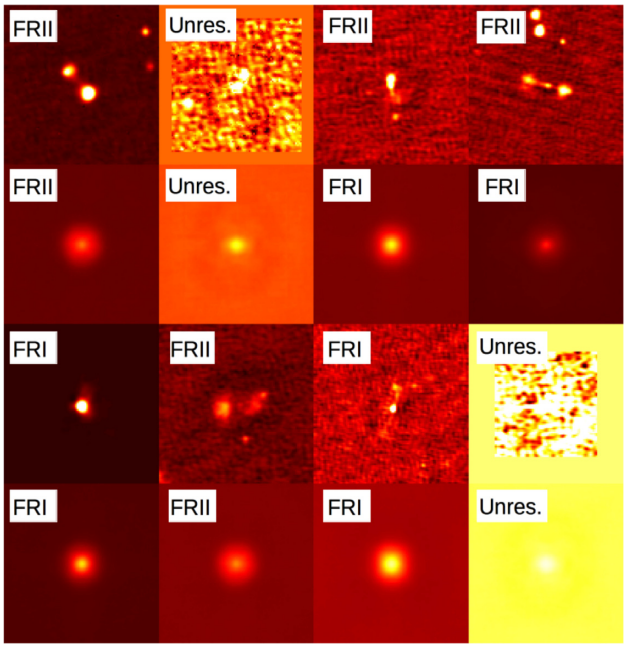

Figure 9. The real and reconstructed images using the default capsule network set-up when training on the original and augmented images, annotated with the corresponding labels. The top row shows the real images, the second row shows the corresponding reconstructions. The third row shows the real images and the final row shows the reconstructions. The decoder always detects that there is an object in the centre of the image, however it is unable to reconstruct the object accurately. Based on the reconstruction, we see that CapsNet is determining class membership based on the characteristics of the sphere in the centre.

'Unresolved' class are represented as concentrated spheres, FRIs are less concentrated, blurrier spheres, and FRIIs are the most diffuse. The inaccuracy of the reconstructions is most likely due to the fact that CapsNet appears to have trouble distinguishing signal from noise. Despite this, the average metrics are still above 89 per cent when training on the original and augmented images, as it does not appear to be necessary to have accurate reconstructions to determine class membership. Overall, the FRII source predictions appear to be the most affected by the noise level and/or potential unassociated emission in the images; since the reconstructions tend to be blurrier spheres with only one component, they become confused with FRIs and FRIIs, as FRIIs can have either both lobes being connected, as well as disconnected.

Figure 10 shows four examples of radio galaxies in which the probabilites are greater than 0.5 across two classes that the CapsNet could therefore not reliably classify. There are a total of 55 out of 600 (9.2 per cent) such cases. Table 10 shows the CapsNet probability vector across the four examples. In Source 1, CapsNet gives similar probabilities between the FRI and FRII classes, which could be because the source is quite faint, therefore it is having trouble extracting the morphology. Source 2 is predicted more confidently as an FRII compared to an Unresolved source, perhaps because it appears as though it has two lobes close together. Sources 3 and 4 are labelled as an FRII, however the CapsNet predicts them more confidently as an FRI compared to an FRII, as it may not detect the lobes.

Similar to what was observed in the ConvNet architectures, the metrics across the FRII class are the poorest. However, after

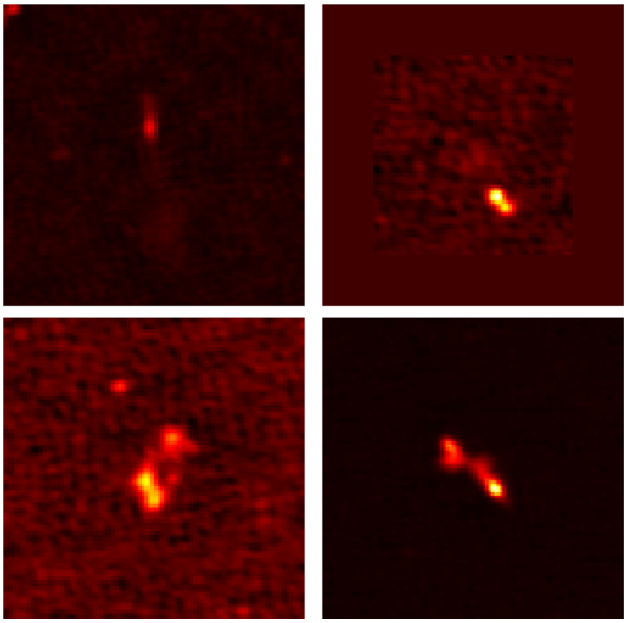

Figure 10 Examples of radio galaxies having probabilitieis greater than 0.5 in more than two classes in the default CapsNet architecture that are also incorrectly predicted. The labels and predictions from left to right, top to bottom are [FRI,Unres.,FRII,FRII] and [FRII,FRII,FRI,FRI], respectively. These sources are labelled as (1, 2, 3, 4) in Table 10.

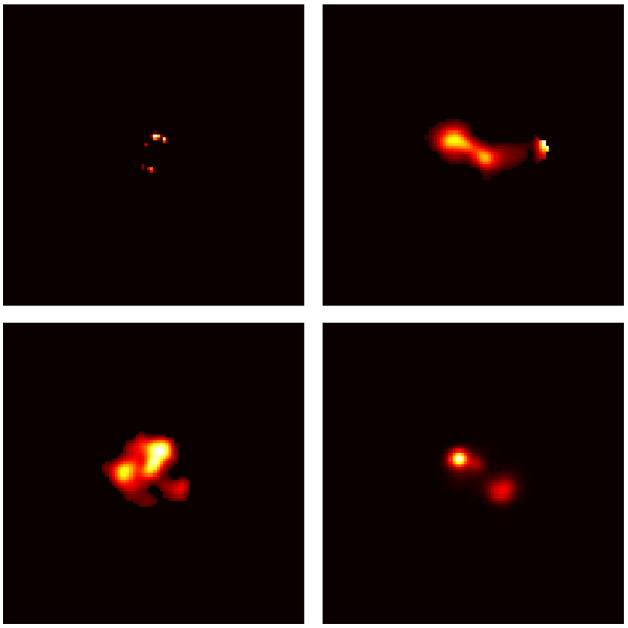

Figure 11. Examples of incorrectly classified radio galaxies from the 4rms sigma-clipped data set using the ConvNet-8 layer architecture. The labels and predictions from left to right, top to bottom are [Unres.,FRII,FRII,FRI] and [FRI,FRI,FRI,FRII], respectively. The top left image appears to have too few pixels to be reliably classified, thus belonging to the 'unresolved' class, however the remaining three may have been misclassified by the automated algorithm.

training with the original and augmented images, the FRII metrics improved the most. The FRII class has the fewest examples of images compared to the other two classes.

Despite the use of image augmentation, it is likely that the number of original training samples available is insufficient to train a capsule network.

5.3 Sigma-clipped images

In order to test whether the CapsNet performance could be improved by removing noise and the occasionally unassociated emission, we used the sigma-clipped images that mask out pixels below 4rms. A flood-filling algorithm and masking techniques have additionally been applied to the data set to identify and connect associated emission (Mingo et al. in preparation). We analyse the results obtained from using the original sigma-clipped images, as well as both the original and augmented images.

The performance of both ConvNets is significantly improved as shown in Tables 5 and 6 (Avg. Recall = 91.9 per cent compared to 88.7 per cent for ConvNet-4, 94.9 per cent compared to 91.2 per cent for ConvNet-8) when using the original sigma-clipped images, compared to using the original FITS files that includes noise and potential unassociated sources. The use of the original sigma-clipped images is significantly worse compared to using the original and augmented FITS images for the ConvNet-4 model (Avg. Recall = 91.9 per cent compared to 93.4 per cent), and is not significantly better for the ConvNet-8 model. The inclusion of augmented images on the sigma-clipped data set appears to benefit the ConvNet-4 model more compared to the ConvNet-8 model.

The performance of CapsNet is significantly improved as shown in Table 7 when using the sigma-clipped original images (Avg. Recall = 91.5 per cent compared to 84.7 per cent with the original FITS images, and compared to 89.9 per cent with the original and augmented FITS images). However, CapsNet still performs worse compared to both ConvNet-4 and ConvNet-8. The use of image augmentation on the sigma-clipped images appears to improve the performance (Avg. Recall = 93.6 per cent compared to 91.5 per cent). The confidence intervals are also generally smaller compared to when the FITS images are used, therefore the performance is slightly more stable.

The use of the sigma-clipped and masked arrays is also significantly better than using the FITS images, when comparing the performance within the original, and the original and augmented data sets, across both ConvNet models and CapsNet models. Therefore, none of the deep learning models can be trained to be completely robust to noise and potentially unassociated emission.

In considering the results of one particular run with the ConvNet-8 model, out of 600 test samples, there are 20 where the predictions do not match the labels. Fig. 11 shows four such examples of images from the 4rms sigma-clipped data set. Upon inspection of all the incorrectly predicted radio galaxies using the ConvNet-8 model, all 12 images that have been labelled as an FRII are predicted to be an FRI. Out of three images labelled as 'Unresolved', two are predicted to be an FRI and one is predicted to be an FRII. The remaining five images labelled as FRI are predicted to be FRIIs. The wrongly classified galaxies mostly appear to have an ambiguous morphology and therefore it could be argued that they are misclassified by the automated algorithm used to label them [see Section 2.2 and Mingo et al. (in preparation)]. For example, the top right and bottom left panels in Fig. 11 do not appear to be a representative examples of an FRII, and the bottom right panel appears more as an FRII, whereas it is labelled as an FRI.

Table 13. The average metrics (in percentages) across each of the classes in the (2) original and augmented LOFAR data set using a 5 convolutional layer model with no intermediate dense layers. Five runs were done in total, using 600 samples in the test set.

Class	Precision	Recall	F1 score	Accuracy
(2)				
Unres.	97.7 ± 0.6	96.9 ± 0.9	97.3 ± 0.3	97.2 ± 0.3
FRI	88.4 ± 1.7	92.8 ± 1.9	90.5 ± 0.1	92.8 ± 0.1
FRII	77.8 ± 2.9	69.0 ± 2.5	73.0 ± 1.3	93.1 ± 0.4
Avg.	91.6 ± 0.2	91.7 ± 0.1	91.6 ± 0.1	95.0 ± 0.1

We note that the larger the proportion of sources that are misclassified by the automated algorithm, the more difficult it will be for the models to learn.

5.4 Additional results

This section summarizes other convolutional and capsule network architectures as well as parameters that were tried. These include transfer learning, the application of early stopping, and comparison of results with similar work.

5.4.1 ConvNet models

We also wanted to test the performance of a simple purely convolutional architecture using five layers (with no intermediate dense layer following the convolutions). The purpose of these dense layers is to help model complex global patterns in the data. The metrics were significantly lower compared to those of both ConvNet models, as shown in Table 13. Therefore, at least one intermediate dense layer could be necessary for optimal performance in convolutional networks. We also tested an architecture using four convolutional with no pooling layers, and found the results to be inferior compared to using the ConvNet-4 model. Therefore, the use of pooling appears to be advantageous in the current data set, perhaps because it allows more degrees of freedom for the morphology within classes.

5.4.2 CapsNet models

Other variations on capsule network models included stacking two convolutional layers instead of one, using 90 per cent training data and 10 per cent testing data, using an ensemble of capsule network models, increasing the number of routing iterations, decreasing the filter size, changing the batch size, adjusting the learning rate, using different activation functions, applying dropout, pooling and using a combination of increased filter sizes together with a more complex decoder, all which resulted in similar or worsened performance metrics. The only possible improvement could be the use of a larger sample of original training images.

5.4.3 Transfer learning

Transfer learning (Pratt, Mostow & Kamm 1991) involves applying the knowledge from one trained neural network to help another learn a related task. In the deep learning context, weights are typically pre-loaded from a network trained on a large data set with many classes to another unseen data set.

We used the Inception ResNet model v2 (Szegedy, Ioffe & Vanhoucke 2016), which combines Inception and Residual network architectures. An inception network consists of a convolutional network using filters of various sizes and pooling within the same layer, and a residual network utilizes skip connections between convolutional layers if the classification accuracy becomes saturated with the subsequent stacking of layers. The Inception ResNet model is trained on the ImageNet data set (Deng et al. 2009), to classify over 14M images into 1000 categories. Although the nature of the ImageNet data set is different to the radio galaxy images, pre-loading weights from a network trained with such a data set is better than initializing the weights from a random distribution.

Table 14. The average metrics (in percentages) across each of the classes in the (2) original and augmented LOFAR data set, for the transfer learning model. Five runs were done in total, using 600 samples in the test set.

Class	Precision	Recall	F1 score	Accuracy
(2)				
Unres.	98.7 ± 0.2	98.3 ± 0.4	98.5 ± 0.2	98.4 ± 0.2
FRI	91.8 ± 0.5	95.0 ± 0.4	93.4 ± 0.2	95.0 ± 0.2
FRII	85.4 ± 1.2	78.7 ± 2.7	81.9 ± 1.2	95.3 ± 0.2
Avg.	94.4 ± 0.2	94.5 ± 0.2	94.4 ± 0.2	96.8 ± 0.1

To use the pre-trained ResNet model in Keras requires images of size of at least 139×139 pixels. As such we padded our images with zeros for 20 pixels along the horizontal and vertical directions, resulting in images of 140×140 pixels.

The pre-trained ResNet model is applied to the LOFAR original and augmented FITS images, to verify whether the classification metrics could be improved from those of our other models. The results in Table 14 show that the classification metrics are not significantly better (Avg. Recall = 94.5 per cent) compared to when training on the same set of images from randomly initialized weights with the ConvNet-8 architecture (Avg. Recall = 94.3 per cent). The metrics are significantly better than for the ConvNet-4 architecture (Avg. Recall = 93.4 per cent). Optimal results are still obtained when using the sigma-clipped data set, where noise and potentially unassociated sources are removed.

We note that the results obtained with transfer learning may be improved if there is a neural network trained on a similar astronomical classification task from which pre-trained weights can be loaded. A successful implementation of transfer learning in classifying optical galaxy morphology is in Dominguez Sanchez et al. (2019), and most recently in radio galaxy morphology classification (Tang, Scaife & Leahy 2019).

The pre-trained network converges faster; ConvNet-4 required 40 epochs of training to reach the optimal validation accuracy as opposed to 30 epochs for the transfer learning model, when averaged over five runs.

5.4.4 Early stopping

We also experimented with applying early stopping in the training of both the Capsnet and ConvNet models. The implementation was such that if the validation accuracy did not improve for 10 subsequent epochs, training was stopped and the metrics on the test set were calculated. However, we found the performance to be the same for the ConvNet model, and worse for the CapsNet model, compared to when training for a pre-defined number of 50 epochs (results not shown). In a work focused on the usage of early stopping, Prechelt (2012) used a mix of more than 1000 training runs across 12 different problems and 24 different architectures

and concluded that slower stopping criteria allow for ≈4 per cent average improvement in generalization, at a cost of around a factor of four longer in training time.

5.4.5 *Recent similar work*

Recently, Katebi et al. (2018) applied a capsule network to classify optical galaxies based on morphology, using the classes of spiral, elliptical, and star/artefact. They find that their capsule network classification accuracy surpasses that of their baseline convolutional network (98.77 per cent versus 96.96 per cent, respectively). The capsule network architecture has over 124M parameters, for a total of 61 578 images. In contrast, our best-performing capsule network uses just over 4M parameters with up to 15 936 images using the original and augmented data set.

We note that the difference in morphology between their classes is starker than in our case. Additionally, the optical images show a much better contrast between object and background, where noise is less prominent. The optical galaxy classifications were crowd-sourced, whereas our labels originated from an automated algorithm which comes with some limitations, as outlined in Section 2.2. The radio emission also produces sparser images compared to the optical galaxy images.

It is difficult to compare their work to ours as the number of images in each of their three classes is unknown. Hence, it is uncertain whether the classification accuracy is the best discriminator to use between the models (Hossin & Sulaiman 2015). Other classification metrics are not provided, such as precision and recall, which may be more powerful in discriminating models. There is also no indication of variability between runs, as well as the degree of overfitting in the networks during training.

6 CONCLUSIONS

This paper explored two deep learning approaches in the classification of radio data from the LoTSS HETDEX field across three classes of radio galaxies: Unresolved sources, FRI, and FRII galaxies. The labels were generated using an automated algorithm, which used a catalogue of sources from the LoTSS DR1 source catalogue with optical IDs and associations (Williams et al. 2019). The radio galaxies belonging to the FRI and FRII classes were additionally cross-checked to eliminate galaxies in which the radio emission is likely to be dominated by star formation (Hardcastle et al. 2019). Despite the classifications being generated using masked images that remove potentially unassociated sources and emission below 4rms from the images, one of our aims was to test how robust our deep learning algorithms could be when such effects were present.

We tested the performance of a four- and eight-layer convolutional neural network (ConvNet-4 and ConvNet-8) against various architectures of capsule networks (CapsNet), using the precision, recall, F1 score, and accuracy, to evaluate the performance of the models. PYTHON code implementing v1.0 of the algorithms can be obtained from github.[8] Automated classifications of LoTSS sources obtained with the algorithms will be presented in a future paper (Mingo et al., in preparation).

The first CapsNet model explored was the default model, a simplified architecture of the original model designed for the MNIST data set, the second used larger filter sizes in the first convolutional layer and Primary capsule layer, and a larger stride in the convolutional layer. The third model used a more complex decoder and a higher loss for the decoder weight. The second and third models were designed to better account for the increased complexity of the data. Four different sets of data were used to train and test the two ConvNets and the variations on CapsNet architectures: (i) using the original FITS images only, (ii) original and augmented FITS images, (iii) the original masked arrays that remove emission below 4rms and potential unassociated sources, and (iv) original and augmented masked 4rms arrays.

We found that the optimal CapsNet performance was obtained when using the default model, in terms of the overall classification metrics.

The results showed that the ConvNet architectures always exceeded the performance of the chosen CapsNet model, and ConvNet-8 always performed better compared to ConvNet-4, most likely because the ConvNet-8 model has twice the number of convolutional layers and parameters as ConvNet-4, therefore it is able to extract higher dimensional features that are particular to each class.

The use of transfer learning on the original and augmented images achieved the same results as ConvNet-8. The performance of all deep learning models was optimized when using the 4rms sigma-clipped numpy array, which is expected as the noise and potential unassociated emission is removed. Some observations of differences in results between using ConvNet and CapsNet architectures and the likely reasons are as follows:

(i) As CapsNet tends to capture and preserve the relative location of features in the images, it is not as successful in distinguishing signal from noise, or dealing with the presence of potentially unassociated emission, as the ConvNet architectures.

(ii) The use of pooling in the ConvNet architectures generally appears to be advantageous in two respects: (a) increased likelihood that noise and potential unassociated sources will be filtered out, (b) allowing more degrees of freedom for variability in morphology within the classes, when the undesirable effects have been removed through use of the 4rms data set.

(iii) The removal of noise and potentially unassociated emission through the use of sigma-clipped and masked arrays improves the performance of both deep learning approaches, when considering the metrics within the original, and original and augmented data sets.

(iv) The use of image augmentation appears to benefit both ConvNets and CapsNet, when using the FITS files, which contain the original radio emission.

The LoTSS survey is the first wide-area survey to contain such faint sources. It is sensitive to a larger range of source evolutionary states, and can also see structure on a wider range of spatial scales due to the combination of well-sampled UV coverage and long baselines. These features result in images having richer, more varied, and sometimes ambiguous morphologies that are more difficult to categorize into distinct classes.

Across both deep learning algorithms, the ‘Unresolved’ class is recovered most successfully, followed by the FRI class. The FRIIs tend to be the least well recovered. Although FRIs display morphological diversity as they can be straight or bent, FRIIs have two peaks of varying distances that may or may not be connected by extended emission with the host galaxy. Therefore, FRIs are more likely to contain a single connected component

[8] https://github.com/vlukic973/RadioGalaxy_Conv_Caps.

whereas FRII can contain either a single or two connected components. There are also fewer examples of FRIIs in the data set compared to FRIs. When we inspected some incorrectly predicted galaxies using the sigma-clipped data set, we found the morphologies to be ambiguous in most cases, as shown in Fig. 11.

Traditional convolutional neural networks generally contain pooling layers in their architecture in order to reduce the number of parameters. However, this can cause the relative locations of features within the image to degrade, which capsule networks are designed to preserve. Our results indicate that for the radio galaxy data in this work, the performance of capsule networks is inferior to that of convolutional neural networks. This could be due to the number of original samples being insufficient to train the capsule network. Another reason may be that since they attempt to preserve the relative location of features, capsule networks appear to interpret noise as signal and introduce extra distortion into the image, as shown in Fig. 5. This aspect has proven to be most detrimental in the recovery of FRII sources, as they are more susceptible to the mingling of signal with noise due to the fact that they are comprised of either one or two components. Additionally, the FRII class contains the fewest examples of images.

In comparison with previous works that use convolutional neural networks to classify radio galaxy morphologies (Aniyan & Thorat 2017; Alhassan et al. 2018; Lukic et al. 2018; Wu et al. 2018), this work explored the use of capsule networks, which are designed to preserve the hierarchical feature information in an image, and finds their performance to be inferior to that of standard convolutional network architectures. The data from the LOFAR LoTSS survey reveals fainter and more detailed emission compared to the data from the surveys which the previous works analysed, providing additional challenges for classification. As such, our findings hold for surveys having a comparable set-up, provided they produce images with similar morphologies and noise profiles.

Based on the current results obtained, it appears that convolutional neural networks still hold as the deep learning technique that should be used for future surveys. They are also faster to train as they use fewer parameters. Capsule networks, in their present form, are generally slower and require further development to be made more robust to noisy real data, however the current performance may be improved by explicitly training them on cleaned data with various examples of morphologies present within each class.

There are several limitations that would need to be overcome to apply these methods to large samples, such as the need for ancillary data to separate star-forming galaxies. The exclusion cannot be performed based purely on the radio galaxy morphology. The classes should also be extended to encompass the hybrid sources, as well as other rare sources such as bent-tailed and double–double sources.

ACKNOWLEDGEMENTS

We thank Huub Rottgering and the anonymous referee for useful comments. VL acknowledges support by the Deutsche Forschungsgemeinschaft (DFG) through grant SFB 676. PNB is grateful for support from STFC via grant ST/R000972/1. This paper is based on data obtained with the International LOFAR Telescope (ILT) under project codes LC2_038 and LC3_008. LOFAR (Haarlem et al. 2013) is the Low-Frequency Array designed and constructed by ASTRON. It has observing, data processing, and data storage facilities in several countries, which are owned by various parties (each with their own funding sources), and which are collectively operated by the ILT foundation under a joint scientific policy. The ILT resources have benefited from the following recent major funding sources: CNRS-INSU, Observatoire de Paris and Universite d'Orleans, France; BMBF, MIWF-NRW, MPG, Germany; Department of Business, Enterprise and Innovation (DBEI), Ireland; NWO, The Netherlands; The Science and Technology Facilities Council (STFC), UK.

REFERENCES

Alger M. J. et al., 2018, MNRAS, 478, 5547
Alhassan W., Taylor A. R., Vaccari M., 2018, MNRAS, 480, 2085
Aniyan A. K., Thorat K., 2017, ApJS, 230, 20
Baldi R. D., Capetti A., Giovannini G., 2015, A&A, 576, A38
Banfield J. K. et al., 2015, MNRAS, 453, 2326
Becker R. H., White R. L., Helfand D. J., 1995, ApJ, 450, 559
Bicknell G. V., 1994, ApJ, 422, 542
Bicknell G. V., 1995, ApJS, 29, 101
Condon J. J., Cotton W. D., Greisen E. W., Yin Q. F., Perley R. A., Taylor G. B., Broderick J. J., 1998, AJ, 115, 1693
Contopoulos I., Gabuzda D., Kylafis N., (eds), 2015, The formation, disruption of black hole jets, Springer
Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L., 2009, in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference. p. 248
Dieleman S., Willett K. W., Dambre J., 2015, MNRAS, 450, 1441
Domínguez Sánchez H., Huertas-Company M., Bernardi M., Tuccillo D., Fischer J. L., 2018, MNRAS, 476, 3661
Fabian A. C., 1999, Proc. Natl. Acad. Sci. USA, 96, 4749
Fanaroff B. L., Riley J. M., 1974, MNRAS, 167, 31 P
Feretti L. 2003, in Bowyer S., Hwang C.-Y., eds, ASP Conf. Ser.Vol. 301, Matter, Energy in Clusters of Galaxies. Astron. Soc. Pac., San Francisco, p. 143
Gopal K., Wiita P. J., 2000, A&A, 363, 507
van Haarlem M. P., , 2013, A&A, 556, A2
Hardcastle M. J. et al., 2019, A&A, 622, A12
Hine R. G., Longair M. S., 1979, MNRAS, 188, 111
Hossin M., Sulaiman M. N., 2015, Int. J. Data Min., Knowl. Manage. Process, 5, 2
Karpathy A., 2016, CS231n Convolutional Neural Networks for Visual Recognition, http://cs231n.github.io/convolutional-networks/
Katebi R., Zhou Y., Chornock R., Bunescu R., 2019, MNRAS, 486, 1539
Kingma D., Ba J., 2014, Adam: A method for stochastic optimization, preprint (arXiv:1412.6980)
Krizhevsky A., Sutskever I., Hinton G. E., 2012, NIPS'12 Proc. 25th International Conference on Neural Information Processing System, 1, 1097
LeCun Y., Bottou L., Bengio Y., Haffner P., 1998, Proceedings of the IEEE, 86, 2278
Lecun Y., Bengio Y., Hinton G., 2015, Nature, 521, 436
Lukic V., Brüggen M., Banfield J. K., Wong O. I., Rudnick L., Norris R. P., Simmons B., 2018, MNRAS, 476, 246
Miraghaei H., Best P. N., 2017, J. Phys.: Conf. Ser., 869, 012078
Nair V., Hinton G. 2010, in Fürnkranz J., Joachims T., eds, Proc. ICML'10, International Conference on Machine Learning. Omnipress, USA, p. 807
Ng A., 2004, in Greiner R., Schuurmans D., eds, Proc. 21st International Conference on Machine Learning. ACM press, New York
Norris R. P. et al., 2011, Publ. Astron. Soc. Aust., 28, 215
Padovani P., 2017, Nat. Astron., 1, 0194

Prandoni I., Seymour N., 2015, in Bourke T. L., Braun R., Fender R., Govoni F., eds, Proc. Advancing Astrophysics with the Square Kilometre Array. Proc. Science, Trieste, Italy, PoS(AASKA14)067
Pratt Y. P., Mostow J., Kamm C. A., 1991, Proceedings of the Ninth National Conference on Artificial Intelligence, Direct transfer of learned information among neural networks. AAAI Press, Menlo Park, CA, p. 584
Prechelt L., 2012, in Montavon G., Orr G. B., Müller K. R., eds, Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, Vol. 7700. Springer, Berlin, Heidelberg
Rumelhart D. E., Hinton G. E., Williams R. J., 1986, Nature, 323, 533
Sabour S., Frosst N., Hinton G. E., 2017, In Advances in Neural Information Processing Systems. Neural Information Processing Systems, NY, USA, p. 3859
Saripalli L., 2012, AJ, 144, 85
Shimwell T. W. et al., 2017, A&A, 598, A104
Shimwell T. W. et al., 2019, A&A, 622, A1
Simonyan K., Zisserman A., 2015, in Bengio Y., LeCun Y., eds, 3rd International Conference on Learning Representations, Published as a conference paper at ICLR
Szegedy C., Ioffe S., Vanhoucke V., Alemi A., 2016, in Singh S., Markovitch S., eds, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), Inception-v4, inception-resnet and the impact of residual connections on learning, AAAI Press, USA
Tang H., Scaife A. M. M., Leahy J. P., 2019, Transfer learning for radio galaxy classification, preprint (arXiv:1903.11921v1)
Willett K. W. et al., 2013, MNRAS, 435, 2835
Williams W. L. et al., 2019, A&A, 622, A2
Wu C. et al., 2019, MNRAS, 482, 1211
Xi E., Bing. S., Jin Y., 2017, preprint (arXiv:1712.03480v1 [stat.ML])

This paper has been typeset from a TEX/LATEX file prepared by the author.