4th URSI AT-RASC, Gran Canaria, 19 – 24 May 2024

Morphological Radio Galaxy Classification with a Fourier Convolutional Neural Network

Barend Edwards*(1), Arno Barnard(1), and Trienko Grobler(2)
(1) Engineering Department, Stellenbosch University, 7600 Stellenbosch, South Africa
(2) Computer Science Department, Stellenbosch University, 7600 Stellenbosch, South Africa

Abstract

Morphological classification of observed radio sources has become fundamental in studies related to the formation and evolution of galaxies and is therefore important to modern astronomy as a whole. Recently, Convolutional Neural Networks (CNNs) have enabled researchers to implement network architectures that are capable of classifying radio galaxies with high accuracies. However, the spatial correlations performed in these networks are computationally expensive. We therefore propose to replace the spatial correlations in a CNN with elementwise multiplications in the frequency domain to reduce the computational complexity, thus reducing the classification time.

1 Introduction

With the anticipation of the Square Kilometre Array (SKA), the field of radio astronomy will see an unprecedented volume of data generation. Average transfer rates to processing facilities are estimated to be 8 Tb/s and 20 Tb/s for SKA-Low and SKA-Mid, respectively. In total, the SKA Observatory expects to archive 300 PB of data per year [1].

Following the Fanaroff-Riley (FR) classification scheme, radio-loud active galactic nuclei (AGNs) can be morphologically classified based on whether they are edge darkened (FRI) or edge brightened (FRII). However, further studies have found that this binary classification scheme is not a sufficient classification scheme due to the rich morphological characteristics that radio galaxies exhibit. Several new classes have been proposed, including bent-tailed, hybrid, and restarting sources [1].

Computer-aided classification of detected radio galaxies from the aforementioned anticipated increase in survey data seems like an inevitable solution for researchers. Since the breakthrough of AlexNet, which won the ImageNet classification challenge in 2012, Convolutional Neural Networks (CNNs) have become the standard solution for computer vision problems [2]. Inspired by the results and architecture of AlexNet, Aniyan & Thorat developed Toothless in 2017 - the first CNN created for the morphological classification of radio galaxies [3]. Their work was pivotal, resulting in a shift towards utilising and creating various CNN architectures for radio galaxy classification [4].

[1] https://www.skao.int/en/explore/big-data

2 Convolutional Neural Networks

This section gives an overview of CNNs by briefly discussing their origin and the working of their three most important layers, namely, fully connected layers, convolutional layers, and pooling layers.

2.1 Artificial Neural Networks

Inspired by the biological architecture of the brain, artificial neural networks (ANNs) are at the centre of deep learning, with the basic building block being the artificial neuron [5]. The output of a single artificial neuron is equal the weighted sum of its inputs and a bias term, as shown in Equation 1.

$$h = \sum_{j=1}^{n} w_j x_j + b \tag{1}$$

The neurophysiological insight gained from different cells found inside a cat's visual cortex led to the development of the Neocognitron - the first introduction of the convolutional neural network [6]. It turned out that the biological processes exhibited by these cells can be modelled as two-dimensional convolution operations.

2.2 Fully Connected Layers

Fully connected layers are formed by stacking layers of artificial neurons, with the input of each neuron in a layer connected to all the outputs of the neurons in a previous layer. Non-linearity is introduced by passing the output of each neuron through an activation function. Equation 2 shows how to calculate the output $h_{l,m}$ of neuron m in layer l, where $\mathbf{h}_{l-1} = [h_{l-1,1}, ..., h_{l-1,n}]$ is the output vector of the neurons in the previous layer (i.e., the inputs of neuron m), $\mathbf{w}_m^l = [w_1, ..., w_n]^T$ is the weight vector applied to the inputs of neuron m, $b_{l,m}$ is the bias applied to neuron m in layer l, and ϕ is the activation function. For all vectors, n is the number of neurons in the previous layer (i.e. layer $l-1$). The case $\mathbf{h}_0$ represents vector of input features $\mathbf{x} = [x_1, ..., x_n]$. The activation function applied to the output layer of the network depends on the task of the network. For a multiclass classification problem, the softmax function is used to normalise the output values into a probability distribution over the different classes [7].

$$h_{l,m} = \phi(\mathbf{h}_{l-1}\mathbf{w}^l_m + b^l_m) \tag{2}$$

2.3 Convolutional Layers

Unlike fully connected layers, the input of neurons in convolutional layers are not connected to every single neuron in the previous layer, but rather only to neurons located within their receptive field. Convolutional layers perform spatial correlations[2] between input feature maps and small kernels of trainable weights. This is done by sliding kernels across input feature maps and computing the sum of the elementwise product between the kernel and the patch of the feature map located within the receptive field of the kernel. Equation 3 shows the spatial correlation between feature map x and kernel w of height f_h and width f_w. Usually the height and width of a kernel is the same value [7, 8].

$$z[i,j] = (x*w)[i,j] = \sum_{u=0}^{f_h-1}\sum_{v=0}^{f_w-1} x[u+i, v+j]\cdot w[u,v] \tag{3}$$

2.4 Pooling Layers

Pooling layers are used to subsample the feature maps, thus reducing the number of parameters that the networks needs to learn. The most common method of pooling is Max Pooling, where the maximum value within a sliding kernel window is passed to the next layer [7].

2.5 Performance Metrics

In this subsection, we outline the machine learning metrics considered in this study. The same metrics as utilized by Becker et al. in his CNN comparison framework are employed in this study [9].

2.5.1 Mean Per Class Accuracy

The overall accuracy of a machine learning model can be a misleading result when working with an imbalanced dataset, as is the case for radio galaxy datasets. We therefore use the mean per class accuracy (MPCA) in this research study. MPCA is calculated by taking the mean of the accuracies of the individual classes.

2.5.2 Model Complexity

Model complexity is measured by the total number of trainable parameters that a model has.

[2]Although we refer to them as convolutional neural networks, it is actually correlation, not convolution, being performed.[7]

2.5.3 Computational Complexity

Computational complexity is measured as the number of floating point operations (FLOPs) used during inference. A FLOP is the addition or multiplication of two floating point numbers.

2.5.4 Loss Ratio

The loss of a model is a measure of the error between the predicted output and the correct output. The loss ratio is obtained by dividing the training and validation losses. A lower loss ratio is indicative of overfitting [10].

2.5.5 Accuracy Ratio

The accuracy ratio is calculated by dividing the training accuracy with the validation accuracy. A model with a higher accuracy ratio is more prone to overfitting.

2.5.6 Rotational Invariance

A CNN is considered rotationally invariant when it can classify an image irrespective of its orientation.

3 Fourier Convolutional Neural Networks

The convolutional layers in a CNN are usually the most computationally expensive elements of the network. For example, the convolutional layers in AlexNet are responsible for almost 90 % of the computations performed [11]. This computational bottleneck leads to the motivation behind Fourier CNNs: employ the correlation theorem to reduce the computational complexity of the model. Equation 4 shows the 2-D discrete correlation theorem, where $[i, j]$ are indices in the spatial domain and $[u, v]$ are indices in the frequency domain. The theorem states that correlation in the spatial domain is equivalent to elementwise multiplication of the Fourier Transforms ($\mathscr{F}$) of the 2-D matricies, given that both matricies are zero-padded to the same size [8]. This theorem introduces a reduction in the computational complexity from $O(N^2K^2)$ for correlation to $O(N^2)$ for elementwise multiplication, where N is the size of the feature map and K the size of the kernel. The replacement of convolutional layers in CNNs with elementwise multiplication layers in the frequency domain has been shown to significantly reduce model complexity with a small reduction in accuracy [11].

$$(x*w)[i,j] \iff (\mathscr{F}\{x\}^* \cdot \mathscr{F}\{w\})[u,v] \tag{4}$$

4 Proposed Fourier CNN

The proposed Fourier CNN (FCNN) is inspired by the architecture and Fourier convolutional layers of CEMNet, a Fourier CNN that achieved competitive accuracies with the Cifar10 dataset [12]. The architectural layout of the FCNN is shown in Table 1. Batch normalization is performed after each pooling layer. The proposed FCNN operates entirely in the frequency domain, which means that no inverse Fourier transforms are performed during classification. It receives complex-valued input data (i.e. already Fourier transformed to the frequency domain). The kernels (with trainable weights) are initialised in the spatial domain. During training, for each epoch, the kernels in the Fourier convolution layers are Fourier transformed before they are elementwise multiplied with the complex feature maps to replicate the function of convolutional layers in a normal CNN. After training is complete, one final Fourier transform of the spatial kernels is performed. This is to store the kernels in the frequency domain so that when the model is loaded to perform classification, no new internal Fourier transforms are performed. The only Fourier transform performed during classification is that of the input data before it is fed to the network.

Table 1. Architecture layout of the proposed Fourier CNN.

Layer	Units	Kernel Size	Activation
FourierConv	32	7	Leaky ReLu
MaxPooling2D		2	
FourierConv	32	5	Leaky ReLu
MaxPooling2D		2	
FourierConv	64	3	Leaky ReLu
MaxPooling2D		2	
FourierConv	128	3	Leaky ReLu
MaxPooling2D		2	
FourierConv	256	3	Leaky ReLu
MaxPooling2D		2	
Flatten			
Dense	512		Leaky ReLu
Dense	256		Leaky ReLu
Dense	256		Leaky ReLu
Dense	3		Softmax

5 Dataset

The dataset used to train and test the models in this study is sampled from the CRUMB (Collected Radiogalaxies Using MiraBest) dataset[3]. It is a machine learning image dataset of 2100 unique Fanaroff-Riley galaxy images of size 150x150 pixels. For this study, only the FRI, FRII, and BENT morphologies were considered. To increase the size of the dataset and test rotational invariance of the CNNs, augmentation was applied by rotating each image 24 times in 15° intervals. The training and testing split per class is shown in Table 2.

6 Methodology

Three different CNNs benchmarked by Becker et al. were chosen for comparison with the proposed FCNN: Toothless [3], FR-DEEP [13], and Radio Galaxy Zoo (RG-Zoo) [14].

[3]CRUMB Dataset: https://zenodo.org/records/8394986

Table 2. Training, validation, and test set break down per class.

Class	Training (Augmented)	Validation (Augmented)	Test (Augmented)
FRI	179 (4296)	47 (1128)	40 (960)
FRII	279 (6696)	84 (2016)	33 (792)
BENT	178 (4272)	29 (696)	48 (1152)
Total	636 (15264)	160 (3840)	121 (2904)

All the models were created and tested in Tensorflow[4] - a Python machine learning library. Each model was trained 15 different times for 20 epochs with a batch size of 32. Nadam optimization was used with a learning rate of 0.001. L2 kernel regularization with a weight decay of 0.01 and dropout was added to all models. For each training instance, a different random seed was applied to all the non-deterministic processes to ensure replicability. The training and validation set was resampled before each training instance, while the testing set remained the same for all instances and all models. The FLOPs each model required to classify a single image during inference was measured with the Tensorflow version 1 profiler[5]. The Fourier transform of the input data is taken into account when measuring the FLOPs of the FCNN.

7 Results

The measured results obtained for all the networks are shown in Figure 1 and Figure 2, with Table 3 providing a summary of the result. In Figure 1, the further a model is from $(1, 1)$, the more prone it is to overfitting. FR-DEEP performs the best in this regard, with a mean distance of 0.1153, while the proposed FCNN performs the worst, with a mean distance of 0.5287.

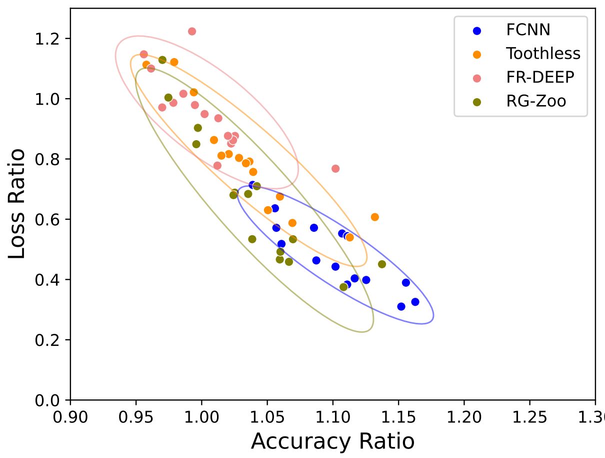

Figure 1. Accuracy Ratio vs Loss Ratio with a 95 % confidence ellipse.

The FCNN is less regularized than the other CNN architec-

[4]https://www.tensorflow.org/
[5]https://github.com/BenEdwards-exe/fourier_rg_cnn

tures, which may contribute to its lower MPCA, shown in Figure 2. However, as expected, the computational complexity of the FCNN is the lowest of the four models. The FCNN requires 299.4× and 2.754× less FLOPS to classify a single image than Toothless and FR-DEEP, respectively.

Table 3. Architecture comparison showing the mean distance from (1,1), the MPCA, and the ratio of FLOPs increase from the FCNN.

Architecture	Mean Distance	MPCA	FLOP Ratio
FCNN	0.5287	84.91	1.0
Toothless	0.2441	90.61	299.4
FR-DEEP	0.1153	89.35	2.754
RG-Zoo	0.3590	90.47	1.664

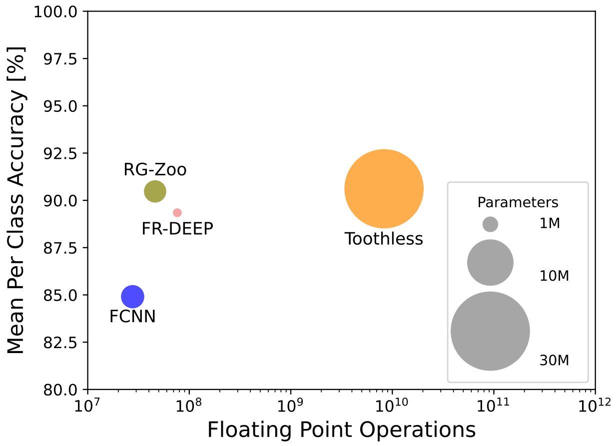

Figure 2. Computational Complexity vs Mean Per Class Accuracy vs Model Complexity.

8 Conclusion

Although the MPCA score is lower for the FCNN, the FLOPs required for inference is significantly less than that of other radio galaxy CNNs. Research into FCNNs as a whole still remain an open problem, with this being the first application of an FCNN for radio galaxy classification.

9 Acknowledgements

The financial assistance of the South African Radio Astronomy Observatory (SARAO) towards this research is hereby acknowledged (www.sarao.co.za).

References

[1] F. A. M. Porter and A. M. M. Scaife, "MiraBest: a data set of morphologically classified radio galaxies for machine learning," *RAS Techniques and Instruments*, vol. 2, no. 1, pp. 293–306, 06 2023.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[3] A. K. Aniyan and K. Thorat, "Classifying radio galaxies with the convolutional neural network," *The Astrophysical Journal Supplement Series*, vol. 230, no. 2, p. 20, 2017.

[4] S. Ndung'u, T. Grobler, S. J. Wijnholds, D. Karastoyanova, and G. Azzopardi, "Advances on the morphological classification of radio galaxies: A review," *New Astronomy Reviews*, vol. 97, p. 101685, 2023.

[5] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[6] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, p. 193–202, 1980.

[7] A. Géron, *Hands-on machine learning with scikit-learn, keras and tensorflow: Concepts, tools and techniques to build Intelligent Systems*, 2nd ed. O'Reilly, 2020.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2018.

[9] B. Becker, M. Vaccari, M. Prescott, and T. Grobler, "CNN architecture comparison for radio galaxy classification," *Monthly Notices of the Royal Astronomical Society*, vol. 503, no. 2, pp. 1828–1846, 02 2021.

[10] A. Roebel, "Dynamic pattern selection for faster learning and controlled generalization of neural networks." 1994.

[11] S. O. Ayat, M. Khalil-Hani, A. A.-H. Ab Rahman, and H. Abdellatef, "Spectral-based convolutional neural network without multiple spatial-frequency domain switchings," *Neurocomputing*, vol. 364, pp. 152–167, 2019.

[12] H. Pan, Y. Chen, X. Niu, W. Zhou, and D. Li, "Learning convolutional neural networks in the frequency domain," *arXiv preprint arXiv:2204.06718*, 2022.

[13] H. Tang, A. M. M. Scaife, and J. P. Leahy, "Transfer learning for radio galaxy classification," *Monthly Notices of the Royal Astronomical Society*, vol. 488, no. 3, pp. 3358–3375, 07 2019.

[14] V. Lukic, M. Brüggen, J. K. Banfield, O. I. Wong, L. Rudnick, R. P. Norris, and B. Simmons, "Radio Galaxy Zoo: compact and extended radio source classification with deep learning," *Monthly Notices of the Royal Astronomical Society*, vol. 476, no. 1, pp. 246–260, 01 2018.