Classification of radio galaxies with trainable COSFIRE filters

Machetho, Stephen; Grobler, Trienko L.; Wijnholds, Stefan J.; Karastoyanova, Dimka; Azzopardi, George

Published in:
Monthly Notices of the Royal Astronomical Society

DOI:
10.1093/mnras/stae821

Publication date:
2024

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **530,** 783–794 (2024) https://doi.org/10.1093/mnras/stae821
Advance Access publication 2024 March 23

Classification of radio galaxies with trainable COSFIRE filters

Steven Ndung'u [1,2★] Trienko Grobler,[2] Stefan J. Wijnholds [2,3] Dimka Karastoyanova[1] and George Azzopardi[1]

[1]*University of Groningen, Information Systems group, Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence, Nijenborgh 9, NL-9712 CP Groningen, the Netherlands*
[2]*University of Stellenbosch, Computer Science, Cnr Banhoek Road & Joubert Street, Stellenbosch 7600, South Africa*
[3]*ASTRON, Oude Hoogeveensedijk 4, NL-7991 PD. Dwingeloo, the Netherlands*

Accepted 2024 March 18. Received 2024 March 15; in original form 2023 November 13

ABSTRACT

Radio galaxies exhibit a rich diversity of morphological characteristics, which make their classification into distinct types a complex challenge. To address this challenge effectively, we introduce an innovative approach for radio galaxy classification using COSFIRE filters. These filters possess the ability to adapt to both the shape and orientation of prototype patterns within images. The COSFIRE approach is explainable, learning-free, rotation-tolerant, efficient, and does not require a large training set. To assess the efficacy of our method, we conducted experiments on a benchmark radio galaxy data set comprising of 1180 training samples and 404 test samples. Notably, our approach achieved an average accuracy rate of 93.36 per cent. This achievement outperforms contemporary deep learning models, and it is the best result ever achieved on this data set. Additionally, COSFIRE filters offer better computational performance, $\sim$20 $\times$ fewer operations than the DenseNet-based competing method (when comparing at the same accuracy). Our findings underscore the effectiveness of the COSFIRE filter-based approach in addressing the complexities associated with radio galaxy classification. This research contributes to advancing the field by offering a robust solution that transcends the orientation challenges intrinsic to radio galaxy observations. Our method is versatile in that it is applicable to various image classification approaches.

Key words: methods: data analysis – methods: statistical – techniques: image processing – galaxies: active – radio continuum: galaxies.

1 INTRODUCTION

Classifying radio galaxies according to their morphology is essential for understanding the physical processes that shape and transform radio galaxies (Hossain et al. 2023; Ndung'u et al. 2023). Automating radio galaxy classification is particularly important for ongoing and upcoming sky surveys: the large volumes of high-resolution data from modern telescopes such as the LOw-Frequency ARray (LOFAR) produce raw data at rates of order TB/s (Rafferty et al. 2013; Shimwell et al. 2022), making manual classification impractical, while public citizen science classification initiatives such as Radio Galaxy Zoo (Banfield et al. 2015) and LOFAR Galaxy Zoo[1] are both time-consuming and rely on the consensus of multiple volunteers. Clearly, these cannot meet the need to label large samples of images [such as the 4.4 million radio source catalogue produced by Shimwell et al. (2022)] promptly, so new methods are required to meet the needs of astronomical population studies. Recently, the development of various innovative machine/deep learning techniques has proven highly effective – achieving remarkable accuracies exceeding 90 per cent (Aniyan & Thorat 2017; Samudre et al. 2022; Ndung'u et al. 2023). These classification models are both robust and generalizable. Despite the limited sample sizes of labelled astronomical data sets (on the order of 10^3), the algorithms have attained generalizability through strategies such as data augmentation and regularization techniques to mitigate overfitting (Aniyan & Thorat 2017; Maslej-Krešňáková, El Bouchefry & Butka 2021; Tang et al. 2022).

Extragalactic radio sources can be differentiated based on the luminosity and the relative positions of the low- and high-brightness regions of the radio sources. Building on the findings of Fanaroff & Riley (1974), their radio galaxy classification system, commonly known as the Fanaroff & Riley (FR) scheme, has gained widespread acceptance. Consequently, radio galaxies are now primarily categorized into four distinct groups, differentiated by their intrinsic morphological characteristics, as outlined by Alhassan, Taylor & Vaccari (2018). The most important two are the Fanaroff & Riley I (FRI) and the Fanaroff & Riley II (FRII) (Fanaroff & Riley 1974). The FRI galaxies have (sometimes diffuse) radio jets and are brighter towards their central core. FRII galaxies, on the other hand, are sources with lobes that are spatially separated from the core. The lobes are the brightest components, outshining both the core and the jets. As a result, the radio emission appears dimmer towards the centre of the galaxy. The third and most common category of radio galaxies is the Compact class, which refers to point-like radio sources (Baldi, Capetti & Giovannini 2015; Baldi, Capetti & Massaro

★ E-mail: s.n.machetho@rug.nl
[1]https://www.zooniverse.org/projects/chrismrp/radio-galaxy-zoo-lofar

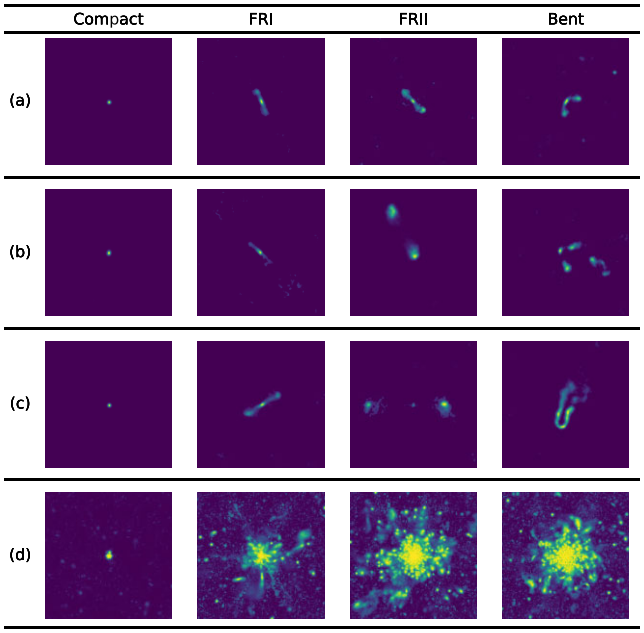


Figure 1. (a)–(c) Sample illustration of the Compact, FRI, FRII and Bent radio sources morphological appearance. (d) Maximum image superposition of all Compact, FRI, FRII, and Bent radio galaxies from the data set used in this paper.

Table 1. The size of the *original* data set distributed across the training, validation, and test categories.

Source catalogue	Type	Total	Training	Validation	Test
Proctor	Bent	508	305	100	103
FR0CAT & CoNFIG	Compact	406	226	80	100
FRICAT & CoNFIG	FRI	389	215	74	100
FRIICAT & CoNFIG	FRII	679	434	144	101
Total		1982	1180	398	404

2018). The Compact sources are similar to FRI, however, they have a higher radio core dominance and radio luminosity (Baldi et al. 2018) – making it a unique class from FRI. The fourth category is Bent, which is composed of radio sources with jets that are bent at an angle, either in a narrow-angled tail (NAT) or a wide-angled tail (WAT) configuration (Rudnick & Owen 1976). The four-class scheme was adopted to maintain consistency and facilitate comparison with other related methods. Radio sources contained within images obtained from highly sensitive, high-resolution telescopes are more resolved, and as such, it is easier to categorize and classify them correctly. This is especially true when it comes to the Compact and Bent radio source classes (Baldi et al. 2018; Harwood, Vernstrom & Stroe 2020). This implies that some sources, given current resolutions, which are labelled as Compact sources, may in actual fact be unresolved FR galaxies. It is also important to mention that radio data set creators have in the past added unresolved sources to the Compact class (Becker et al. 2021). In Fig. 1 we illustrate a few examples of each of the four classes as well as superimposed variants created by utilizing all images from the data set described in Table 1. In addition to the difficulties related to their shape characteristics, the classification of FRI, FRII, and Bent galaxies is further exacerbated by the fact that the galaxies in the data set have differing orientations. Upon visual inspection, it becomes apparent that each category exhibits a discernibly distinct brightness distribution. Nonetheless, it is worth noting that the FRII and Bent classes share substantial similarities, thus making it difficult to discern between the two classes and a challenging task to differentiate them.

In this paper, we propose a lightweight paradigm that involves trainable COSFIRE (Combination of Shifted Filter Responses) filters (Azzopardi & Petkov 2012a; Azzopardi, Greco & Vento 2016a). This approach is efficient, learning-free, rotation-tolerant, explainable, and does not require a large training set. The implementation of a COSFIRE-based classification pipeline is relatively easy and straightforward from a conceptual standpoint as described in Section 4. It involves the configuration of COSFIRE filters (Azzopardi & Petkov 2012a) whose selectivity of each filter is automatically determined from the shape properties of a single training example. The objective is to set up multiple filters, whose combined responses generate a feature signature for the type of galaxy present in an image. This approach is analogous to how visual cells in the mammalian brain are thought to encode visual information, a concept known as population coding (Pasupathy & Connor 1999, 2002). COSFIRE filters have been applied in various computer vision tasks: retina image analysis (Azzopardi et al. 2015; Strisciuglio, Azzopardi & Petkov 2019; Ramachandran et al. 2020), crack detection (Strisciuglio, Azzopardi & Petkov 2017), traffic sign recognition (Gecer, Azzopardi & Petkov 2017), keyword spotting in handwritten manuscripts (Azzopardi & Petkov 2014), machine vision (Azzopardi et al. 2016b), delineation of apposing mitochondria in electron microscopy images (Aswath et al. 2023), gender recognition from face images (Azzopardi et al. 2016a), contour detection in images with natural scenes (Azzopardi & Petkov 2012b), and handwritten digit classification (Azzopardi & Petkov 2013). In this work, COSFIRE filters are configured to extract the hyperlocal geometric arrangements that uniquely describe the patterns of radio sources (in terms of blobs) in a given image.

The rest of the paper is structured as follows. Section 2 presents the current state-of-the-art approaches. Section 3 describes the data set used in this study. Section 4 describes the proposed COSFIRE-based paradigm. Section 5 presents the evaluation criteria used to assess the performance of our approach. Section 6 describes the experiments and the results obtained. Section 7 provides a discussion of the results obtained in relation to the relevant work. Finally, we draw conclusions in Section 8.

2 RELATED WORKS

End-to-end learning, particularly through convolutional neural networks (CNNs), has dominated the field in recent years. In particular, Aniyan & Thorat (2017), used the AlexNet architecture (Krizhevsky, Sutskever & Hinton 2017), calling the trained model Toothless,[2] to achieve accuracies of 91 per cent, 75 per cent, and 95 per cent for the FRI, FRII, and Bent-tailed morphologies, respectively (on a data set they assembled from various catalogues). Subsequently, notable incremental breakthroughs have been made in the applications of deep learning to the field of radio astronomy, ranging from shallow CNN architectures (Lukic et al. 2019) to deep and complex architectures such as DenseNet (Huang et al. 2017; Samudre et al. 2022). Furthermore, other noteworthy advancements within the field are: model-centric strategies such as group-equivariant CNNs (G-CNNs) (Scaife & Porter 2021) to support equivariance translations on various isometries of radio galaxies and multidomain multibranch CNNs (Tang et al. 2022) to allow models to learn jointly from various

[2]https://github.com/ratt-ru/toothless

survey inputs; data-centric approaches such as data augmentation (Ma et al. 2019; Maslej-Krešňáková et al. 2021; Kummer et al. 2022; Slijepcevic et al. 2022); transfer learning (Lukic et al. 2019; Tang, Scaife & Leahy 2019), and *N*-shot learning (Samudre et al. 2022) (algorithms that have been developed to optimally utilize limited amounts of supervised information while mitigating obstacles such as overfitting) to overcome the limited availability of annotated data sets in radio astronomy. These models have been shown to perform competitively, providing promising alternatives to prior models such as the one by Aniyan & Thorat (2017), which showed signs of overfitting.

Feature-based pipelines offer a more traditional approach, involving handcrafted features, and have been explored for the morphological classification of Fanaroff-Riley (FR) radio galaxies (Becker et al. 2021; Ntwaetsile & Geach 2021; Sadeghi, Javaherian & Miraghaei 2021; Darya et al. 2023). These approaches are not end-to-end as they decouple the feature description and the classification model. They rely heavily on the use of handcrafted feature descriptors, such as Haralick features derived from the Grey Level Co-occurrence Matrix (Haralick, Shanmugam & Dinstein 1973; Ntwaetsile & Geach 2021). Gradient boosting methods (Friedman 2002), including XGBoost (Chen & Guestrin 2016), LightGBM (Ke et al. 2017), and CatBoost (Dorogush et al. 2018), are also utilized in this context. This emphasis on deliberate feature engineering has been shown to produce promising results (Sadeghi et al. 2021), particularly in smaller data sets. For instance, Darya et al. (2023) demonstrated that these feature-based methods can be as effective as CNN-based deep learning models when applied to data sets of around 10 000 images or fewer.

CNNs are regarded as the state-of-the-art in various image classification applications (Aniyan & Thorat 2017; Lukic et al. 2019; Scaife & Porter 2021; Tang et al. 2022). However, they require large amounts of training data and are susceptible to overfitting when trained with small data sets, which is the case in radio astronomy. Moreover, the high computational demands of deep architectures for training and applying CNNs often require GPUs, which can be costly and limit their applicability in resource-limited settings. Additionally, CNN-based models lack insufficient intrinsic robustness to rotations. To address rotational variations in radio sources, multiple approaches have been taken. One approach is to utilize group-equivariant CNNs, where the network is designed to capture the diverse orientation information of a given input galaxy in encoded form (Scaife & Porter 2021). Another method involves augmenting the training data by applying rotations to the training samples, enabling the CNNs to learn different orientations of the classes. Furthermore, a pre-processing step can be employed to standardize the rotation of all radio sources. This may be achieved by using principal component analysis to align the galaxies' principal components with the coordinate system's axes, effectively normalizing their orientations (Brand et al. 2023).

As evident from this literature review, numerous challenges remain, including the need for efficient (computationally inexpensive) and rotationally invariant methods. In this work we address these limitations with the proposed COSFIRE filter approach.

3 DATA

The data set of radio galaxies used in this paper was compiled and processed by Samudre et al. (2022). It was constructed by selecting well-resolved radio galaxies from multiple catalogues: Proctor catalogue (Proctor 2011) for the Bent radio galaxies; FR0CAT catalogue (Baldi et al. 2018) and Combined NVSS–FIRST galaxies catalogue (CoNFIG) (Gendre & Wall 2008; Gendre, Best & Wall 2010) for Compact radio galaxies; FRICAT catalogue (Capetti, Massaro & Baldi 2017a) and CoNFIG catalogue for FRI radio galaxies and finally FRIICAT catalogue (Capetti, Massaro & Baldi 2017b) and CoNFIG catalogue for FRII radio galaxies. In this paper, we utilize this data set to perform comparative analyses, specifically in relation to the work conducted by Samudre et al. (2022).

Table 2. The size of the *balanced* data set distributed across the training, validation, and test categories.

Source catalogue	Type	Total	Training	Validation	Test
Proctor	Bent	680	433	144	103
FR0CAT & CoNFIG	Compact	675	431	144	100
FRICAT & CoNFIG	FRI	674	430	144	100
FRIICAT & CoNFIG	FRII	679	434	144	101
Total		2708	1728	576	404

The initial data set is composed of the following classes: Compact (406 samples), Bent (508 samples), FRI (389 samples), and FRII (679 samples). These samples are further divided into training, validation, and testing sets as shown in Table 1. According to Samudre et al. (2022), the original data set's underrepresented classes were balanced by adding randomly duplicated samples to the training and validation data sets. Table 2 depicts the distribution of the balanced data set.

The images were pre-processed by utilizing sigma-clipping with a threshold of 3σ (Aniyan & Thorat 2017). This technique involves eliminating or discarding pixels that have background noise levels above or below 3 standard deviations from the mean (Aniyan & Thorat 2017).

Although more recent catalogues exist, as described by Ndung'u et al. (2023), the latest catalogues, such as the LOFAR Two-metre Sky Survey – Data Release I & II (Shimwell et al. 2019, 2022), are not annotated according to the Fanaroff & Riley classification scheme. Consequently, they were not utilized in this study, as their inclusion would hinder direct comparison of our results with other related methods. Instead, they are labelled by automated tools such as the Python Blob Detector and Source-Finder (PyBDSF) (Mohan & Rafferty 2015) that categorize astronomical sources into three types: 'S' for single isolated sources modelled with one Gaussian, 'C' for sources that are within a group but can be individually modelled with a single Gaussian, and 'M' for extended sources that need multiple Gaussians for accurate modelling. This system aids in the efficient identification and analysis of space emissions.

4 METHODS

This section gives a comprehensive description of the methodology that we propose for radio galaxy image classification with the COSFIRE filter approach. We explain the process of radio source blob detection, the configuration of COSFIRE filters with rotation invariance, the formation of feature descriptors, and the utilization of these descriptors for classification.

4.1 Blob detection

Blob detection is a technique used for identifying points or regions in an image that exhibit a sudden change in intensity (areas that are either brighter or darker than the surrounding areas), known as a 'blob'. This approach enables the identification of regions that may correspond to objects or structures of interest within an image; in our case radio source(s). In computer vision, one of the commonly used blob detectors is based on the Laplacian of Gaussian (LoG) (Wang,

Lopez-Molina & De Baets 2017, 2020). The LoG is the second-order derivative of the Gaussian function, which we denote by $G(x, y)$:

$$G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (1)$$

where σ is the standard derivation. The Laplacian of Gaussian is typically estimated by a Difference-of-Gaussian (DoG) function, which is separable, and thus convolving it with a 2D image is much more computationally efficient. Given two Gaussian functions, G_1 and G_2 with their respective standard deviations σ_1 and σ_2, the Gaussians are defined as:

$$G_1(x, y; \sigma_1) = \frac{1}{2\pi\sigma_1^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_1^2}\right) \quad (2)$$

$$G_2(x, y; \sigma_2) = \frac{1}{2\pi\sigma_2^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_2^2}\right) \quad (3)$$

The DoG function is obtained by subtracting one Gaussian function from another, each having a different standard deviation:

$$\mathrm{DoG}(x, y; \sigma_1, \sigma_2) = G_1(x, y; \sigma_1) - G_2(x, y; \sigma_2) \quad (4)$$

For a point (x, y) and an image I with intensity distribution $I(x', y')$, we calculate the response $c_{\sigma_1,\sigma_2}(x, y)$ of a DoG filter with a kernel function $\mathrm{DoG}_{\sigma_1,\sigma_2}(x - x', y - y')$ by convolution:

$$c_{\sigma_1,\sigma_2}(x, y) = \mathrm{ReLU}(I \star \mathrm{DoG}_{\sigma_1,\sigma_2})_{t_1}, \quad (5)$$

where ReLU is a rectification linear unit, as adapted from Nair & Hinton (2010), and it serves as an activation function that assigns zero to all values below the given threshold t_1 [see equation (6)], and $\star$ represents convolution,

$$\mathrm{ReLU}(x) = \begin{cases} 0 & \text{if } x < t_1 \\ x & \text{if } x \geq t_1 \end{cases} \quad (6)$$

When $\sigma_1 < \sigma_2$ we refer to the resulting DoG function as a centre-on DoG or DoG^+ for brevity, with the central region (radio source) exhibiting a positive response and the surrounding (background) exhibiting a negative response. Conversely, when $\sigma_2 < \sigma_1$ the configuration results in a centre-off DoG function, which we denote by DoG^-. In this work, we adopt the approach used by Azzopardi et al. (2015) and always set the smaller standard deviation to be half of the larger standard deviation.

The centre-on DoG filter highlights bright blobs on a dark background and is more sensitive to intensity increases at the centre of the blob. This filter is effective in detecting and highlighting areas of the image with radio sources (blobs) relative to the background noise and is less sensitive to edges or sharp changes in intensity. On the other hand, the centre-off DoG filter highlights dark blobs on a bright background and is more sensitive to intensity decreases at the centre of the blob. This filter responds well to corners and edges and is less sensitive to objects or regions of uniform intensity. Therefore, both centre-on and centre-off DoG filters are used to detect blobs or regions of interest in an image, see Fig. 2 for an illustration on Compact, FRI, FRII, and Bent radio sources. The DoG filter is a type of band-pass filter that can eliminate high-frequency components that represent noise as well as some low-frequency components that represent homogenous areas.

4.2 COSFIRE filter configuration

A COSFIRE filter is automatically configured by examining the shape properties of a given prototype pattern of interest in an image, which ultimately determines its selectivity. The process of configuration can be summarized in three main steps: *convolve-ReLU-keypoint detection*. The first two steps involve the convolution of centre-on and centre-off DoG filters followed by ReLU as described above. Finally, keypoint detection requires the determination of local maximum thresholded DoG responses along a set of concentric circles around a point of interest. The point of interest is a location in the image that characterizes the radio source(s), and consequently it is the location where the configured COSFIRE filter is expected to respond maximally. For this application, we use the centre of the image as the point of interest, but in principle, any location can be used for this purpose. The number and radii of the concentric circles along with the threshold t_1 used by the ReLU function are hyperparameters of the COSFIRE approach. A keypoint denoted by k, which is identified as a local maximum of a DoG filter along a concentric circle, is characterized by a four-element tuple: $(\sigma_k, \delta_k, \rho_k, \phi_k)$. Here, σ_k and δ_k indicate the standard deviation of the outer Gaussian function of the DoG function and its polarity (centre-on or centre-off) that achieved the highest response in the polar coordinates with radius ρ_k at an angle of ϕ_k radians with respect to the given point of interest. We denote by C_f a COSFIRE filter, which is represented as a list of such tuples:

$$C_f = \{(\sigma_k, \delta_k, \rho_k, \phi_k) \mid k = 1, \ldots, n\}, \quad (7)$$

where n refers to the number of DoG responses considered in C_f, which plays a crucial role in the selectivity and generalization of the COSFIRE filter. Typically, selectivity increases and generalization decreases with increasing value of n (i.e. number of keypoints).

Fig. 3 demonstrates the automatic configuration of a COSFIRE filter, showcasing the superposition of centre-on and centre-off DoG response maps derived from a Bent class image in the training set, the one shown in Fig. 2. The example uses two concentric circles and the resulting COSFIRE filter is a set of ten tuples describing the ten keypoints indicated in Fig. 3(b). The keypoints are identified at the positions along the circles where the DoG responses reach local maxima.

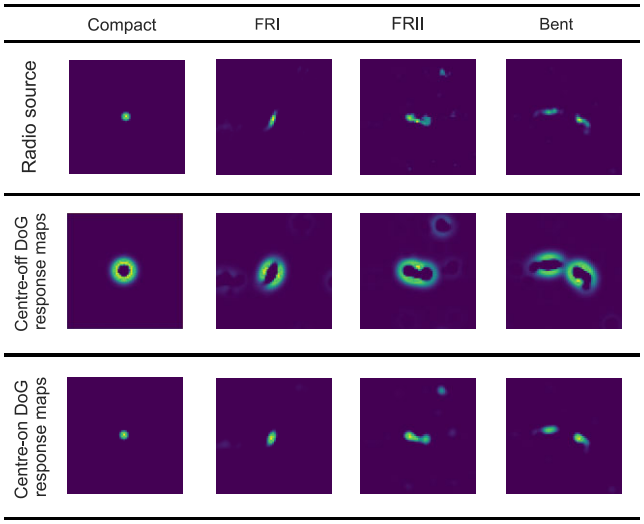
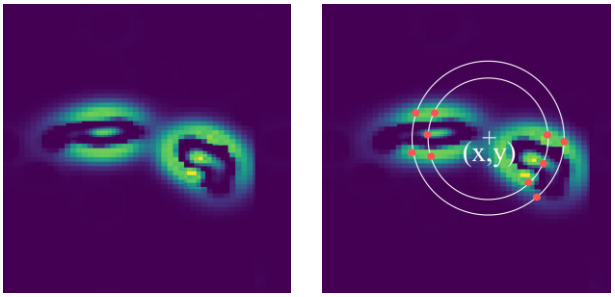


Figure 2. Centre-on and centre-off DoG filter response maps obtained with convolution on example images from the Compact, FRI, FRII, and Bent classes of radio sources, utilizing a standard deviation σ of 3 for the outer Gaussian function. The images are of size 150 × 150 pixels.



(a) (b)

Figure 3. Example of keypoint detection for the configuration of a COSFIRE filter. (a) A Bent class image of the superposition of a centre-on and a centre-off DoG response maps. This is the same example used in the last column of Fig. 2. (b) The ten dots indicate the ten keypoints that are identified in locations that exhibit local maximum DoG values along the given two radii with respect to the centre (x, y).

4.3 COSFIRE filter response

The response of a COSFIRE filter C_f for a given (x, y) location is computed by combining the responses of the DoG functions whose scale σ_k, polarity δ_k, and position (ρ_k, ϕ_k) with respect to (x, y) are indicated in the set C_f. The process of computing the response can be summarized in five main steps: *convolve-ReLU-blur-shift-combine*. The *convolve* step refers to the convolution of centre-on and centre-off DoG filters followed by the *ReLU* operation that sets to zero all values below the given threshold t_1. These are the same two steps that were required for the configuration described above. Then, in order to allow for some tolerance in the preferred positions of the involved DoG responses, we *blur* each DoG response with a Gaussian function $G_{\sigma'}(x', y')$, whose standard deviation σ' is a linear function of the distance ρ_k: $\sigma' = \lfloor \sigma'_0 + \alpha\rho_k \rfloor$, with σ'_0 and α being hyperparameters. Moreover, we *shift* the blurred responses in the direction opposite of the polar coordinates such that all DoG responses of interest meet at the same (x, y) location. Finally, the *combine* step is implemented as suggested by Azzopardi & Petkov (2012a), where a COSFIRE filter response, which we denote by r_{C_f}, is obtained by the geometric mean function that combines all blurred and shifted thresholded DoG responses:

$$r_{C_f}(x, y) = \left(\prod_{k=1}^{n} z_{\sigma_k, \delta_k, \rho_k, \phi_k}(x, y) \right)^{\frac{1}{n}}, \tag{8}$$

where

$$z_{\sigma_k, \delta_k, \rho_k, \phi_k}(x, y) = \max_{x', y'} \{ c_{\sigma_k}(x - \Delta x_k - x', y - \Delta y_k - y') G_{\sigma'}(x', y') \} \tag{9}$$

is the combined blurred and shifted DoG filter response map of tuple k. The shifting operation displaces each blurred DoG response of interest to the support centre of the COSFIRE filter. The shift vector is denoted by $(\Delta x_k, \Delta y_k)$, where $\Delta x_k = -\rho_k \cos\phi_k$ and $\Delta y_k = -\rho_k \sin\phi_k$. Additionally, $-3\sigma' \leq x', y' \leq 3\sigma'$.

4.4 Tolerance to rotations

Tolerance to rotations can be attained by configuring multiple COSFIRE filters using rotated versions of a single prototype pattern. An effective approach to achieve this involves defining new filters by modifying the parameters of an existing COSFIRE filter. For instance, consider a COSFIRE filter denoted as C_{f_ψ}, designed to be selective for the same underlying pattern that was employed to configure the original COSFIRE filter C_f, but rotated by ψ radians. This new filter is defined as follows:

$$C_{f_\psi} = \{(\sigma_k, \delta_k, \rho_k, \phi_k + \psi) \mid \forall\, (\sigma_k, \delta_k, \rho_k, \phi_k) \in C_f\} \tag{10}$$

The rotation-tolerant response of a COSFIRE filter $\hat{r}_{C_f}$ is then achieved by taking the maximum response across all COSFIRE filters selective for the same pattern at 12 equally-spaced orientations:

$$\hat{r}_{C_f}(x, y) = \max \left\{ r_{C_{f_\psi}(x,y)} \,\forall\, \psi \in \{0, \pi/6, \ldots, 11\pi/6\} \right\} \tag{11}$$

4.5 COSFIRE descriptor

For a given image, a COSFIRE descriptor, which we denote by D and define below, is generated by applying all COSFIRE filters in rotation-tolerant mode and extracting the maximum value from each filter, regardless of its location. Consequently, for a set of λ filters, the resulting description of a given image is represented by a λ-dimensional vector.

$$D = \left[\max_{x,y} \left\{ \hat{r}_{C_{f_1}}(x, y) \right\}, \ldots, \max_{x,y} \left\{ \hat{r}_{C_{f_\lambda}}(x, y) \right\} \right] \tag{12}$$

This concept is inspired by neurophysiology research, which suggests that the shape representation of a stimulus is based on the combined activity of a group of shape-selective neurons in area V4 (Wielaard et al. 2001; Azzopardi & Petkov 2012b; Weiner & Ghose 2015). V4 cells are neurons in the visual cortex that are involved in form perception, recognizing objects and their features such as shape (Weiner & Ghose 2015).

4.6 COSFIRE descriptor pre-processing

The only pre-processing step done to the COSFIRE descriptors is L2 normalization (Dai et al. 2018). L2 normalization is performed by dividing the original vector by its magnitude, where the magnitude is calculated as the square root of the sum of the squares of the vector's elements. This process effectively scales the vector to have a unit length in Euclidean space. The key advantage of L2 normalization is its ability to make the descriptors robust to variations in illumination and contrast. By ensuring that the magnitude of feature vectors is consistent, L2 normalization enables more accurate and reliable comparisons between features extracted from images with varying conditions, such as different contrast or brightness levels, thereby improving the performance of image classification and recognition tasks.

4.7 Classification model

The SVM model, introduced by Cortes & Vapnik (1995), has been selected for the classification of COSFIRE-based image descriptors due to its capacity to handle high-dimensional data, manage outliers, and achieve robust generalization. The training of the SVM model is conducted using the COSFIRE descriptors extracted from a training set that encompasses four distinct classes of radio galaxies. Given the imbalance in class distribution of the training set, as highlighted in Table 1, we utilize a bagging (bootstrap aggregating) approach to train an ensemble of classification models. This method involves creating balanced subsets of the training data for each model in the ensemble. Employing balanced subsets effectively reduces biases toward the majority classes, enhancing the overall performance and fairness of the classification. The inherent advantage of the bagging

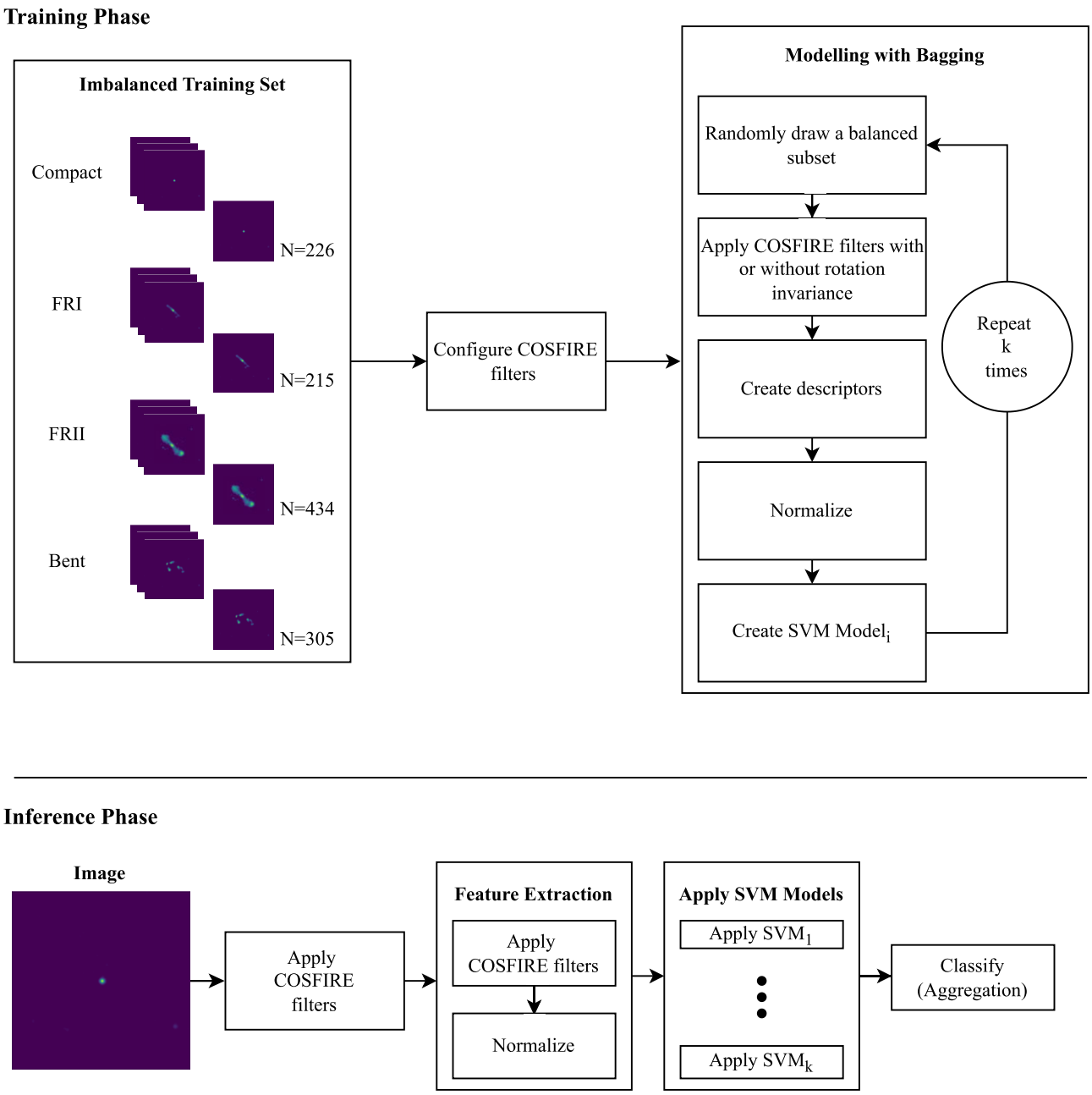


Figure 4. A graphic representation of the proposed COSFIRE-based pipeline for radio galaxy classification for both training and inference phases. The training phase involves configuring COSFIRE filters, extracting descriptors, and training an ensemble of classifiers using a bagging approach. Specifically, we create $k = 10$ classifiers (where i represents an individual classifier in this series). These models are applied in the inference stage and the label of the given image is determined by decision fusion.

strategy, as proposed by Breiman (1996), not only addresses the skewness in class distribution but also improves the generalization of the ensemble to unseen data by aggregating diverse models trained on varied representations of the data. Fig. 4 illustrates the entire process of training and inference. In the training phase, we employ a resampling technique with replacement to randomly select balanced subsets of images from each class. To achieve this balance, we calculate two-thirds of the size of the majority class (in this case, 434), which results in 290 images. By selecting 290 images from each class, we also ensure that the resulting balanced subset is roughly the same size as the original imbalanced training set. Using these balanced subsets, we then train a set of ten different SVM models. In the inference phase, we use the ten SVM models to perform classification by aggregating their predictions on the test data set. In practice, the subset size and the number of classifiers are two hyperparameters of the bagging approach, which can be fine-tuned by cross validation. For the individual SVM models, we use default parameter values,[3] including a linear kernel and a cost parameter C

[3]https://github.com/cjlin1/libsvm/blob/master/matlab/svmtrain.c

set to 1. The fine-tuning of the SVM hyperparameters is beyond the scope of the proposed COSFIRE paradigm.

5 PERFORMANCE METRICS FOR EVALUATION

5.1 Accuracy

As in prior research on morphological classification (Lukic et al. 2019; Samudre et al. 2022), we evaluate the performance of the proposed COSFIRE approach following the widely adopted accuracy metric, which computes the proportion of correctly classified images out of the total number of images tested.

5.2 Floating point operations

Floating point operations (FLOPs) are a measure of computer performance that indicates the number of floating-point operations that a processor executes to complete a task. This measurement is commonly used in scientific programs/applications that heavily depend on floating-point (FP) calculations such as CNNs. In Section 6.3, we explain in detail how the computational complexity of our approach is measured in terms of the number of FLOPs required. We then benchmark this against the DenseNet161 classification model, which achieved the highest performance on the same data set we employ, as reported by Samudre et al. (2022). This comparison aims to contextualize our approach's efficiency relative to the top-performing model in the existing literature.

FLOPs estimation for both algorithms is performed during the inference phase rather than the training phase of the classification process. This phase entails only the forward pass of input data through the model to produce classifications, thereby focusing on the computational cost of executing the already trained model.

6 EXPERIMENTS AND RESULTS

In this section, we present and analyse the experimental results obtained in our research on radio galaxy classification. We also compare the performance and computational complexities of our method with other existing approaches in radio galaxy classification.

6.1 Performance

We conducted a series of experiments to evaluate the performance of the proposed trainable COSFIRE filter approach. As detailed in Section 4, our pipeline involves the automatic configuration of COSFIRE filters and their application to training, validation, and test data sets for each distinct class (Bent, Compact, FRI, and FRII). Our primary objective was to attain the highest classification accuracy for distinguishing among the four radio source classes. Through this rigorous evaluation process, we conducted an in-depth exploration of the trainable COSFIRE filters' performance and their potential in the realm of radio source classification.

We used the validation data set to determine the hyperparameters of the COSFIRE filter configuration (σ, the set of radii P used to configure COSFIRE filters, and t_1) and application (σ_0', and α). In our experimentation, we explored three values per hyperparameter as shown in Table 3. This resulted in a total of 243 unique parameter sets. For every unique set, we configured up to 100 COSFIRE filters for each class, leading to a total of 400 COSFIRE filters. The filters were configured by selecting random images from the training set. To compensate for the randomness of the image selection, we executed three experiments with the same set of hyperparameters and then took the average results across the three experiments. These experiments resulted in a 243×400 matrix of accuracy rates. We then identified the maximum accuracy rate for each row (i.e. for each set of hyperparameters). Importantly, multiple sets of hyperparameters yielded very close accuracies. Therefore, to account for all the hyperparameters generating similar results, we performed a right-tailed student t-test statistic (Marshall & Jonker 2011) between the row with the global maximum accuracy rate and all the other rows. A right-tailed student t-test statistic is used in hypothesis testing to determine if there is a statistically significant increase in the mean performance of one set of hyperparameters compared to another set or a specified benchmark. In our analysis, we discovered that 26 sets of hyperparameters demonstrated performance levels that were not significantly lower than the performance of the best-performing set, referred to as the 'global maximum'. This implies that while these 26 sets did not necessarily outperform the global maximum, their performance was comparable, indicating no significant statistical difference in terms of inferiority. The average maximum accuracy of the experiments with these 26 sets of hyperparameters was achieved with 93 COSFIRE filters per class on the validation set.

Table 3. Search space for hyperparameter tuning of COSFIRE filters. This table lists the sets of values for each hyperparameter that were explored to determine the optimal configuration of COSFIRE filters.

Parameter name	Values
σ	$\{5, 6, 7\}$
P	$\left\{ \begin{array}{l} \{0, 5, 10, 15, 20\}, \\ \{0, 5, 10, 15, 20, 25\}, \\ \{0, 5, 10, 15, 20, 25, 30\} \end{array} \right\}$
t_1	$\{0.05, 0.1, 0.15\}$
σ_0'	$\{0.5, 0.75, 1\}$
α	$\{0.1, 0.15, 0.2\}$

Subsequently, we employed the COSFIRE filters in conjunction with the corresponding classifiers, which were configured using 26 distinct sets of hyperparameters determined previously on the validation set, to the test data. The outcomes of this process are graphically represented in Fig. 5. This figure includes two principal plots illustrating the variation in accuracy rates as a function of the number of COSFIRE filters used. One plot delineates the performance when the filters operate in a rotation-invariant manner, while the other depicts the scenario without rotation invariance. Each plot is the average across the results obtained by the experiments using the 26 sets of hyperparameters, with the grey shading indicating the standard deviations. With 93 filters per class (as determined from the validation data) and rotation invariance, the average accuracy on the test set is 93.36 ± 0.57 per cent (with the minimum being 92.24 per cent and the maximum being 94.31 per cent), whereas without rotation invariance, the same filters yielded an average accuracy of 82.25 ± 2.06 per cent. This performance trend is visually depicted in Fig. 5, highlighting that as the number of filters increases, the disparity in performance between including and excluding rotation becomes more pronounced. Additionally, the figure shows that the model performance is more stable with rotation invariance than without it. The latter has high variability even with more COSFIRE filters.

6.2 Comparison with previous works

The most direct comparison to our study is the research that used DenseNet-161 (Huang et al. 2017) and Siamese networks (Koch

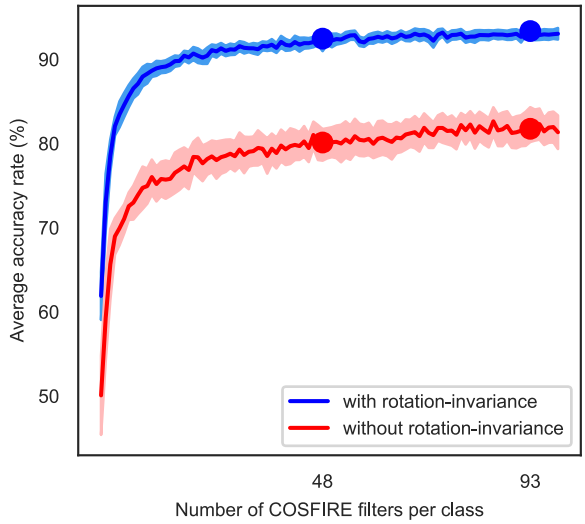


Figure 5. Accuracy rate achieved on the original test set (see Table 1) as a function of the number of COSFIRE filters used. The plots demonstrate how the average accuracy rate over the experiments with the determined 26 sets of hyperparameters varies based on the number of COSFIRE filters used, considering both cases of rotation invariance and without rotation invariance. Specifically, employing 48 filters for each class approximates the 92 per cent accuracy level of DenseNet161, as reported by Samudre et al. (2022). Using 93 filters per class, which led to the maximum average accuracy on the validation set, we achieve an accuracy of 93.36 per cent on the test set. The four spots indicate the accuracies achieved with 48 and 93 filters per class, both with and without rotation invariance.

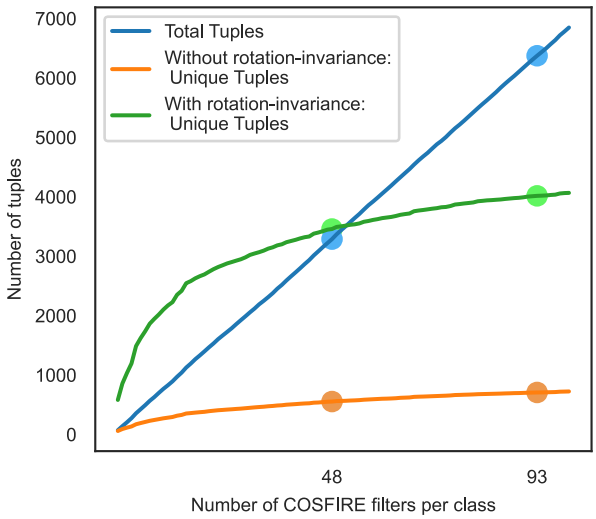


Figure 6. Number of tuples as a function of the number of COSFIRE filters used per class. The total number of tuples increases linearly with the increase in the number of COSFIRE filters per class, but the number of unique tuples increases sublinearly, since most tuples are shared among all COSFIRE filters and stored in the memory, hence requiring less computations. The total number of tuples required by 48 filters is very close to the number required by 93 filters.

et al. 2015) methodologies conducted by Samudre et al. (2022) on the same data. In their work, they conducted their experiments on two versions of the data: the original and the balanced data set as presented in Table 1 and Table 2, respectively. According to their reported results, Siamese networks and DenseNet-161 achieved the highest accuracies of 71.1 ± 0.40 per cent and 91.2 ± 0.60 per cent, respectively, on the original data set. On the other hand, Siamese networks achieved 73.9 ± 0.50 per cent and DenseNet-161 achieved 92.1 ± 0.40 per cent accuracy on the balanced data set. Notably, our COSFIRE approach achieves an accuracy of 93.36 ± 0.57 per cent without data augmentation or any further image pre-processing. This is more than 16 per cent reduction in the classification error rate compared to the DenseNet-161 model (Samudre et al. 2022). The significant reduction in the error rate is mainly attributed to the rotation invariance properties of the COSFIRE approach. We did not perform runs on the balanced data set (Table 2) since this would not generate new insights, as the data set was augmented through image duplication of the minority classes.

6.3 Computational complexity

6.3.1 COSFIRE filter approach

The FLOPs associated with the single convolution required by the COSFIRE filter approach are computationally inexpensive due to the exploitation of the separability property of Gaussian filters. For the separability of the Gaussian filter, equation (13), the 2D Gaussian can be written as the multiplication of two functions: one that depends on x and the other that depends on y. In this case, the two functions are the same and they are both 1D Gaussian. This reduces the number of operations required to apply the filter to an image. For example, applying a 5 × 5 filter to an image requires 25 multiplications per pixel, but applying two 1D filters of length 5 requires only 10 multiplications per pixel. Because the DoG function combines two Gaussian functions linearly, the separability property is also preserved in the DoG function.

$$G(x,y;\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)\times\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{y^2}{2\sigma^2}\right) \quad (13)$$

Table 4 provides a breakdown of the FLOPs that are required by the COSFIRE methodology at the inference stage. In that table, we use the following set of hyperparameters as an example: $\sigma = 5$, $P \in \{0, 5, 10, 15, 20\}$, $t_1 = 0.1$, $\sigma_0 = 0.5$, and $\alpha = 0.1$. The FLOPs computations are mainly on the processes of convolve-ReLU-blur-shift-combine. For a given image of 150 × 150 pixels and 400 COSFIRE filters, the total estimated number of FLOPs is ∼1.5 GFLOPs. Also, in order to optimize the computational efficiency in a system using 4800 COSFIRE filters (i.e. 100 filters for each of the four classes applied in 12 orientations) that collectively involve 64 284 tuples, we employ the following strategy:

(i) **Elimination of redundancy**: Recognizing that numerous tuples are repeated across these filters, we first isolate every distinct tuple from the entire set. In fact, the number of tuples increases sublinearly with the increase in the number of COSFIRE filters used, Fig. 6.

(ii) **Computation and storage**: We calculate the response map for each unique tuple only once. This response map is essentially a modified version of the centre-on or centre-off DoG response map, subjected to blurring and shifting. We then store each computed response map in a hash table for quick retrieval, avoiding redundant computations.

(iii) **Configuration sharing**: Upon further analysis, we find that the configurations of many COSFIRE filters share identical pairs of tuples.

Table 4. COSFIRE filter FLOPs estimation at inference stage. The FLOPs calculation in the columns *FLOPs basic* and *FLOPs* (computation with pre-computed response maps of pairs of tuples) are based on the number of tuples obtained when using a single image for inference. As an example here is a breakdown of FLOPs at each stage for the specific set of hyperparameters: $\sigma = 5$, $P \in \{0, 5, 10, 15, 20\}$, $t_1 = 0.1$, $\sigma'_0 = 0.5$, and $\alpha = 0.1$. In the *FLOPs* column, the symbol – represents the equivalent value from the *FLOPs basic* column.

Step	Formula	FLOPs basic	FLOPs
Centre-On DoG sigma: Convolve the $m \times n(150 \times 150)$ image with a DoG kernel whose size is $s = 6\sigma + 1$ (here $\sigma = 5$). The expression a combines the product and summation operations, and the result is multiplied by 2 to account for the vertical and horizontal separable filters of the DoG functions.	$a = 2(mn(2s - 1))$	2 745 000	–
Centre-off DoG (same sigma): Centre-off is simply the negative of the centre-on response map.	$b = mn$	22 500	–
ReLU of centre-on DoG map: Assigns all values below t_1 to zero across the centre-on responses using ReLU activation function.	$c = mn$	22 500	–
ReLU of centre-off DoG map: Same as above but for the centre-off response map.	$d = mn$	22 500	–
Blurring: Similar to a, convolve the $m \times n$ (150×150) image with a DoG kernel of size: $\sigma'_k = \lfloor \sigma_0 + \alpha\rho_k \rfloor$. The expression a is multiplied by two to account for centre-on and centre-off DoG response maps. The extent of blurring depends on the radii: 0, 5, 10, 15, and 20. The kernel size of the blurring function for the k-th radius is calculated as $s_k = 6\hat{\sigma}_k + 1$: $s_0 = 6(1) + 1 = 7$, $s_5 = 7$, $s_{10} = 13$, $s_{15} = 13$, $s_{20} = 19$.	$\hat{e}_{(\rho_k=0)} = 2(2(mn(2(7) - 1)))$ $\hat{e}_{(\rho_k=5)} = 2(2(mn(2(7) - 1)))$ $\hat{e}_{(\rho_k=10)} = 2(2(mn(2(13) - 1)))$ $\hat{e}_{(\rho_k=15)} = 2(2(mn(2(13) - 1)))$ $\hat{e}_{(\rho_k=20)} = 2(2(mn(2(19) - 1)))$ $e = \sum_{i \in \{0,5,10,15,20\}} \hat{e}_{(\rho_k=i)}$	1 170 000 1 170 000 2 250 000 2 250 000 3 330 000	– – – – –
Shifting: Shift the two DoG response maps as many times as the number of all unique tuples t_p across all COSFIRE filters. In our experiments, the t_p was 3187 (with rotation invariance).	$f = mnt_p$	71 707 500	–
Multiplication: Compute the responses of all COSFIRE filters by multiplying the corresponding shifted response maps. The number of rotations is denoted by $n_r = 12$. The variable $n_d = 4800$ represents the number of COSFIRE filters, which here is set to 400 (100 per class) and multiplied by the total number of rotations ($n_r = 12$). T_p represents the total number of tuples across all the COSFIRE filters ($T_p = 64, 284$). The set of total tuples configured considering rotation invariance have repetitions that do not need to be re-computed. In this step, we pair the tuples that appear more than once among the 64 284 tuples. Then, we only compute and store in the memory a single pair of the shared tuples (duplicates) to save on computations. Therefore, the FLOPS are calculated from the pre-computed response maps of pairs of tuples and those of unique tuples.	$g = mnT_p - mnn_d$	1 338 390 000	937 305 300
Hashkey: Before applying the multiplication operation, the shifted response map must first be retrieved from a hashtable residing in memory. The keys of this hashtable are determined by multiplying the first four prime numbers raised to given σ, δ, ρ and ϕ: $h_k = 2^\sigma 3^\delta 5^\rho 7^\phi$. The retrieval of each tuple response map from memory is therefore 7 FLOPS.	$h = T_p h_k n_r.$	449 988	–
Taking the root for geometric mean: Geometric mean calculation operations for each response map of all n_d COSFIRE filters.	$i = mnn_d n_r$	108 000 000	–
Descriptor formation: Determination of a $n_{dp} = 400$-element descriptor by taking the maximum value of each of the n_{dp} response maps.	$j = mnn_{dp}$	9 000 000	–
Decision fusion of 10 SVM classifiers: SVM calculations depend on the number of decision function hyperplanes (n_{hp}) given the number of classes under classification. This experiment uses the one-vs-one ('ovo') decision function approach, which means that $n_{hp} = z(z - 1)/2$. In this case, $z = 4$ and hence $n_{hp} = 6$. Each hyperplane separates one class from the rest of the classes. The number of FLOPs is therefore based on the dot products needed for one SVM between the given feature vector of size n_{dp} and the n_{hp} hyperplanes. Since the dot product involves multiplications and additions then one SVM takes $2n_{dp}n_{hp}$ FLOPs. Having 10 SVM, this is repeated 10 times and finally, we choose the maximum which takes another 9 FLOPs.	$k = 10(2n_{dp}n_{hp}) + 9$	48 009	–
Total FLOPs	$a + b + c + d + e + f + g + h + i + j + k$	1 540 825 497	1 139 740 797

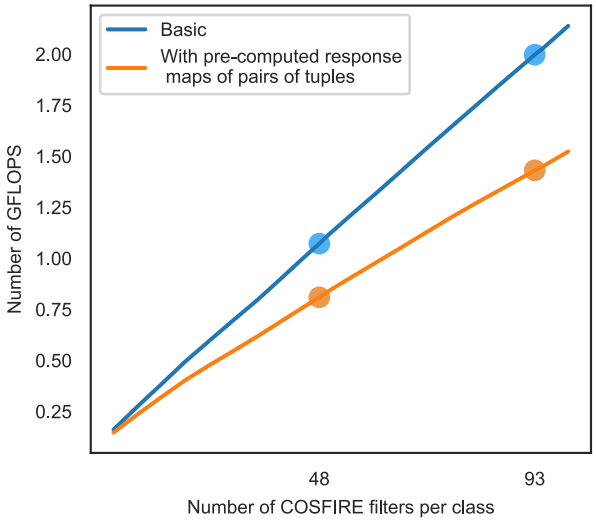


Figure 7. The number of FLOPs with and without the pre-computation of response maps of pairs of tuples against the number of COSFIRE filters per class.

(iv) **Pairwise response maps**: We compile a list of these common tuple pairs and pre-calculate the combined response map for each pair by performing a multiplication of the individual response maps.

(v) **HashTable for pairs**: The resulting combined response maps of tuple pairs are also stored in a hash table. This ensures that the response for any pair that is used by more than one filter can be quickly fetched without recalculating it.

By implementing this approach, we make the process more efficient, as we avoid unnecessary recalculations for both individual tuples and their pairs, which significantly reduce the computational load and speeds up the overall filtering process. Using this efficient technique, the total number of FLOPs required is reduced to ∼1.1 GFLOPs; i.e. 26 per cent reduction in the FLOPs (comparison of FLOPs basic and FLOPs column values in Table 4).

Similarly, conducting three trials with 26 sets of optimal hyperparameters from the validation data set, we noted a significant decrease in FLOPs, as depicted in Fig. 7, corresponding to an increasing number of COSFIRE filters. In fact, with reference to Fig. 7 and Fig. 5 we demonstrate that we can achieve very high performance with fewer filters using the COSFIRE approach. Using just 48 filters for each class yields an accuracy rate of 92.46 ± 0.76 per cent, with a computational cost of 0.8 GFLOPs–approximately 20 times less than the DenseNet161 model that achieves a similar result.

6.3.2 *Convolutional neural networks*

Calculating the FLOPs in a convolutional neural network involves determining the number of arithmetic operations performed during the forward pass. This includes the multiplications and additions between the input data elements and the trainable convolution kernels. The FLOPs computation takes into account factors such as the input size, kernel size, number of filters, and the presence of padding, dilation, and stride. Padding is the process of adding extra pixels around the input image to maintain size post-convolution. Dilation refers to spreading out kernel elements to cover a larger input area without increasing filter size. Stride is the step size of the filter across the image, affecting downsampling. For a comprehensive definition and explanation of padding, dilation, stride, as well as a general introduction to CNNs, we refer the reader to O'Shea & Nash (2015). A CNN convolutional operation with padding set to 0, dilation set to 1, and stride set to 1, for simplicity (Freire et al. 2022), can be summarized as follows:

$$y^f_{x,y} = \phi\left(\sum_{i=1}^{n_H}\sum_{j=1}^{n_W}\sum_{k=1}^{n_C} K^f_{i,j,k} \cdot I_{x+i-1,y+j-1,k} + b^f\right) \tag{14}$$

In equation (14), the output of a specific convolutional layer is denoted by $y^f_{x,y}$, which represents a feature map. Here, the superscript f indicates the filter index, while the subscripts x, y, and k represent the row, column, and channel indices of the output feature map, respectively. The trainable convolution kernel is denoted by $K^f_{i,j,k}$, and it performs element-wise multiplications with the corresponding input elements of image I of dimensions $n_H \times n_W \times n_C$. To meet the input requirements of a DenseNet model, the original images are resized from 150 × 150 to 224 × 224 × 3 pixels. The bias term b^f is added to the sum, and the resulting value undergoes the activation function $\phi(\cdot)$ to generate the output feature map element $y^f_{x,y}$. This equation embodies the convolution operation in a CNN, where each output feature map element is computed by convolving the corresponding region of the input feature maps with trainable filters and introducing non-linearity through the activation function.

The output size of the CNN at each layer is important when calculating the FLOPs of the CNN architecture used. It is given by,

$$\text{OutputSize} = \left[\frac{n_h + 2p - k}{\text{s}} + 1\right], \tag{15}$$

where n_h is the image input size, p is padding, k is the kernel size, and s is the stride. The number of FLOPs can then be estimated as:

$$\text{FLOPs}_{\text{CNN}} = n_f n_i n_k \cdot \text{OutputSize}, \tag{16}$$

where n_f is the number of filters. In each sliding window, there are n_i n_k multiplications and the sliding window process needs to be repeated as many times as the OutputSize. This entire procedure is then repeated for all n_f filters. The estimated number of FLOPs of the DenseNet-161 model is 15.6 GFLOPs based on the workflow by Sovrasov (2023); roughly GMACs = 0.5 ∗ GFLOPs.[4] Importantly, the 15.6 GFLOPs calculations include the operations involving the dense and activation layers in the DenseNet architecture. Other commonly used CNN benchmarking architectures, such as AlexNet, VGG16Net, and ResNet50 for an image of size 224 × 224, require 1.44, 31.04, and 8.26 GFLOPs, respectively, for processing an image. Original images would need to be resized from 150 × 150 to 224 × 224 × 3 pixels to align with the input specifications of these models. The COSFIRE approach, besides its superior computational efficiency, offers a distinct advantage in terms of flexibility. Notably, the COSFIRE method stands out for its adaptable architecture, allowing users to customize the number of COSFIRE filters employed according to their specific needs. This flexibility is a significant benefit, enabling efficient computation without sacrificing performance. In practice, utilizing merely 48 COSFIRE filters per class results in an architecture that uses only 0.8 GFLOPs and approximates the same accuracy of 92 per cent achieved by DenseNet161 (see Fig. 7).

7 DISCUSSION

The proposed trainable COSFIRE filter approach, which is designed to capture radio galaxies at different orientations, achieved better

[4] https://github.com/sovrasov/flops-counter.pytorch

results than the CNN-based approach for radio galaxy classification on the same data set. This suggests that the trainable COSFIRE filter approach can capture more relevant salient features of the radio galaxy images than the CNN-based approach. The shape of radio galaxies that we observe depends on how their jets are oriented in the plane of the sky; making rotation invariance crucial for robust and accurate galaxy classification. Therefore, for effective model generalization, it is imperative that a classification model remains invariant to rotations, ensuring consistent predictions irrespective of the input's orientation.

In other similar studies, multiple approaches exist for addressing the challenge of rotation invariance using different data sets in radio astronomy (Becker et al. 2021; Scaife & Porter 2021; Brand et al. 2023). Brand et al. (2023) showed that classification performance is improved when the orientations of training galaxies are normalized as opposed to when no such attempt was made to address rotational variations. They achieved an accuracy of 96.10 per cent. In Scaife & Porter (2021), group-equivariant CNNs were applied to capture various orientations of a given input galaxy, achieving an optimal accuracy of 96.56 per cent. Moreover, the widely adopted data augmentation pre-processing step in machine model development is another approach that also addresses this challenge of radio galaxy equivariance. Becker et al. (2021) applied this approach and achieved a mean per class accuracy of 71.75 per cent. While data augmentation is a useful tool, especially for small data sets, it increases the cost of training the classification model. In addition, it requires domain knowledge to develop a good augmentation strategy, which considers all possible equivariant transformations given a galaxy image. The rotation invariance in the COSFIRE approach is intrinsic to the method and does not require data augmentation. This makes the approach more robust, versatile, and completely data-driven, with no domain expertise required, and hence highly adaptable to various computer vision applications. Importantly, our results align with the studies discussed above that explore various approaches to enhance robustness against different orientations of radio galaxies. However, those approaches use different data sets and processing steps, making them not directly comparable. This provides an avenue for further research, where a common data set and consistent processing steps would be applied to assess performance and energy requirements for a scalable, robust, and rotation-invariant radio astronomical classification model.

CNN-based networks are computationally expensive. For instance, the forward pass of DenseNet161 consumes ∼15.6 GFLOPs. On the other hand, the COSFIRE filter approach demonstrates efficiency by utilizing a substantially lower number of FLOPs (Fig. 7), resulting in significantly lower demand for computational resources. The workflow of the COSFIRE approach is also more flexible than conventional CNN architectures. The number of filters used is a hyperparameter of this paradigm, which is in contrast to the fixed number of filters used in its counterpart. Our approach, based on the COSFIRE filter, is a novel algorithm for the classification of radio galaxies and comes with a low computational cost.

8 CONCLUSION

In this work, we introduce a novel descriptor based on trainable COSFIRE filters for radio galaxy classification. We combine this descriptor with an SVM classifier using a linear kernel and achieve better performance compared to the only previous work on the data set proposed by Samudre et al. (2022).[5] The previous study utilized a DenseNet161 transfer learning-based pre-trained network and few-shot learning-based Siamese networks, as detailed in Samudre et al. (2022). Our technique is computationally inexpensive (∼20 × lower cost with the same accuracy as the DenseNet161 model), rotation invariant, free from data augmentation and does not rely on domain expert knowledge. These features make our technique not only effective (in terms of accuracy) but also efficient (in terms of FLOPs) for radio galaxy image classification tasks. We achieved a better accuracy of 93.36 per cent compared to 92.10 per cent which was achieved by the DenseNet161 model.

This work contributes to the field of radio astronomy by providing an alternative technique for identifying and analysing radio sources. As the next-generation telescopes (such as LOFAR, MeerKAT, and SKA) produce high-resolution images of the sky, our future work will explore the effectiveness of our methodology on these new data and assess the possibility of cross-survey predictions.

ACKNOWLEDGEMENTS

Part of this work is supported by the Foundation for Dutch Scientific Research Institutes. This work is based on the research supported in part by the National Research Foundation of South Africa (grant numbers 119488 and CSRP2204224243).

We thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Hábrók high performance computing cluster.

DATA AVAILABILITY

The data set is available at: https://github.com/kiryteo/RG_Classification_code.

REFERENCES

Alhassan W., Taylor A., Vaccari M., 2018, MNRAS, 480, 2085
Aniyan A., Thorat K., 2017, ApJS, 230, 20
Aswath A., Alsahaf A., Westenbrink B. D., Giepmans B. N., Azzopardi G., 2023, in International Conference on Computer Analysis of Images and Patterns. p. 87
Azzopardi G., Petkov N., 2012a, IEEE Trans. Pattern Anal. Mach. Intell., 35, 490
Azzopardi G., Petkov N., 2012b, Biol. Cybernet., 106, 177
Azzopardi G., Petkov N., 2013, in Computer Analysis of Images and Patterns: 15th International Conference, CAIP 2013, York, UK, August 27–29, 2013, Proceedings, Part II 15. p. 9
Azzopardi G., Petkov N., 2014, Front. Comput. Neurosci., 8
Azzopardi G., Strisciuglio N., Vento M., Petkov N., 2015, Medical Image Anal., 19, 46
Azzopardi G., Greco A., Vento M., 2016a, in 13th IEEE international conference on advanced video and signal based surveillance (AVSS). p. 235
Azzopardi G., Fernández-Robles L., Alegre E., Petkov N., 2016b, in 2016 23rd International Conference on Pattern Recognition (ICPR). p. 3356
Baldi R. D., Capetti A., Giovannini G., 2015, A&A, 576, A38
Baldi R. D., Capetti A., Massaro F., 2018, A&A, 609, A1
Banfield J. K. et al., 2015, MNRAS, 453, 2326
Becker B., Vaccari M., Prescott M., Grobler T., 2021, MNRAS, 503, 1828
Brand K., Grobler T. L., Kleynhans W., Vaccari M., Prescott M., Becker B., 2023, MNRAS, 522, 292
Breiman L., 1996, Mach. Learn., 24, 123

[5]https://github.com/kiryteo/RG_Classification_code

Capetti A., Massaro F., Baldi R. D., 2017a, A&A, 598, A49
Capetti A., Massaro F., Baldi R., 2017b, A&A, 601, A81
Chen T., Guestrin C., 2016, in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. p. 785
Cortes C., Vapnik V., 1995, Mach. Learn., 20, 273
Dai Z., Chen W., Huang X., Li B., Zhu L., He L., Guan Y., Zhang H., 2018, in 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO). p. 144
Darya. A. M. et al., 2023, preprint(arXiv:2304.12729)
Dorogush A. V. et al., 2018, preprint(arXiv:1810.11363)
Fanaroff B. L., Riley J. M., 1974, MNRAS, 167, 31P
Freire P. J., Srivallapanondh S., Napoli A., Prilepsky J. E., Turitsyn S. K., 2022, preprint (arXiv:2206.12191)
Friedman J. H., 2002, Comput. Stat. Data Anal., 38, 367
Gecer B., Azzopardi G., Petkov N., 2017, Image Vision Comput., 57, 165
Gendre M. A., Wall J. V., 2008, MNRAS, 390, 819
Gendre M. A., Best P. N., Wall J. V., 2010, MNRAS, 404, 1719
Haralick R. M., Shanmugam K., Dinstein I. H., 1973, IEEE Trans. Syst. Man Cybernet., SMC-3, 610
Harwood J. J., Vernstrom T., Stroe A., 2020, MNRAS, 491, 803
Hossain M. S., Roy S., Asad K., Momen A., Ali A. A., Amin M. A., Rahman A. M., 2023, Proc. Comput. Sci., 222, 601
Huang G., Liu Z., Van Der Maaten L., Weinberger K. Q., 2017, in Proceedings of the IEEE conference on computer vision and pattern recognition. p. 4700
Ke et al., 2017, Advances in neural information processing systems, 30
Koch G. et al., 2015, in ICML deep learning workshop.
Krizhevsky A., Sutskever I., Hinton G. E., 2017, Commun. ACM, 60, 84
Kummer J. et al., 2022, preprint (arXiv:2206.15131)
Lukic V. et al., 2019, MNRAS, 487, 1729
Ma Z. et al., 2019, ApJS, 240, 34
Marshall G., Jonker L., 2011, Radiography, 17, e1
Maslej-Krešňáková V., El Bouchefry K., Butka P., 2021, MNRAS, 505, 1464
Mohan N., Rafferty D., 2015, Astrophysics Source Code Library, record ascl–1502
Nair V., Hinton G. E., 2010, in Proceedings of the 27th international conference on machine learning (ICML-10). p. 807
Ndung'u S., Grobler T., Wijnholds S. J., Karastoyanova D., Azzopardi G., 2023, New Astron. Rev., 97, 101685
Ntwaetsile K., Geach J. E., 2021, MNRAS, 502, 3417
O'Shea K., Nash R., 2015, preprint (arXiv:1511.08458)
Pasupathy A., Connor C. E., 1999, J. Neurophysiol., 82, 2490
Pasupathy A., Connor C., 2002, Neuroscience, 5, 1252
Proctor D. D., 2011, ApJS, 194, 31
Rafferty D. et al., 2013, A&A, 556, A2
Ramachandran S., Strisciuglio N., Vinekar A., John R., Azzopardi G., 2020, Neur. Comp. Appl., 32, 12453
Rudnick L., Owen F. N., 1976, ApJ, 203, L107
Sadeghi M., Javaherian M., Miraghaei H., 2021, AJ, 161, 94
Samudre A., George L. T., Bansal M., Wadadekar Y., 2022, MNRAS, 509, 2269
Scaife A. M., Porter F., 2021, MNRAS, 503, 2369
Shimwell T. W. et al., 2019, A&A, 622, A1
Shimwell T. et al., 2022, A&A, 659, A1
Slijepcevic I. V. et al., 2022, MNRAS, 514, 2599
Sovrasov V., 2018-2023, ptflops: a flops counting tool for neural networks in pytorch framework. Available at: https://github.com/sovrasov/flops-counter.pytorch
Strisciuglio N., Azzopardi G., Petkov N., 2017, in Felsberg M., Heyden A., Krüger N., eds, Computer Analysis of Images and Patterns. Springer International Publishing, Cham, p. 108
Strisciuglio N., Azzopardi G., Petkov N., 2019, IEEE Trans. Image Process., 28, 5852
Tang H., Scaife A. M., Leahy J., 2019, MNRAS, 488, 3358
Tang H., Scaife A., Wong O., Shabala S., 2022, MNRAS, 510, 4504
Wang G., Lopez-Molina C., De Baets B., 2017, in Proceedings of the IEEE International Conference on Computer Vision. p. 4817
Wang G., Lopez-Molina C., De Baets B., 2020, Digital Signal Process., 96, 102592
Weiner K. F., Ghose G. M., 2015, J. Neurophysiol., 113, 3021
Wielaard D. J., Shelley M., Mclaughlin D., Shapley R., 2001, J. Neurosci., 21, 14

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