Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **518,** 3123–3136 (2023)
Advance Access publication 2022 November 16

https://doi.org/10.1093/mnras/stac3336

Photometric identification of compact galaxies, stars, and quasars using multiple neural networks

Siddharth Chaini,[1] Atharva Bagul,[1★] Anish Deshpande,[2] Rishi Gondkar,[3] Kaushal Sharma,[4] M. Vivek[5] and Ajit Kembhavi[6]

[1] *Department of Physics, Indian Institute of Science Education and Research, Bhopal 462066, India*
[2] *Department of Computer Science Engineering, Indian Institute of Technology, Bombay 400076, India*
[3] *Department of Computer Science, Pune Institute of Computer Technology, Pune 411043, India*
[4] *Millennium Institute of Astrophysics (MAS), Nuncio Monseñor Sótero Sanz 100, Providencia, Santiago, Chile*
[5] *Indian Institute of Astrophysics, Koramangala, Bengaluru 560034, India*
[6] *Inter University Centre for Astronomy and Astrophysics (IUCAA), Pune 411007, India*

Accepted 2022 November 11. Received 2022 November 11; in original form 2022 June 18

ABSTRACT

We present `MargNet`, a deep learning-based classifier for identifying stars, quasars, and compact galaxies using photometric parameters and images from the Sloan Digital Sky Survey Data Release 16 catalogue. `MargNet` consists of a combination of convolutional neural network and artificial neural network architectures. Using a carefully curated data set consisting of 240 000 compact objects and an additional 150 000 faint objects, the machine learns classification directly from the data, minimizing the need for human intervention. `MargNet` is the first classifier focusing exclusively on compact galaxies and performs better than other methods to classify compact galaxies from stars and quasars, even at fainter magnitudes. This model and feature engineering in such deep learning architectures will provide greater success in identifying objects in the ongoing and upcoming surveys, such as Dark Energy Survey and images from the Vera C. Rubin Observatory.

Key words: methods: data analysis – techniques: photometric – software: data analysis – stars: general – galaxies: general – quasars: general.

1 INTRODUCTION

Large-scale surveys like the Sloan Digital Sky Survey (SDSS; York et al. 2000), the Dark Energy Survey (DES; Flaugher 2005), the Zwicky Transient Facility (ZTF; Bellm 2014), and the Subaru Prime Focus Camera (Miyazaki et al. 2002) collect photometric and spectroscopic data on millions of objects like stars, galaxies, and quasars. The forthcoming Vera C. Rubin Observatory and its Legacy Survey of Space and Time (LSST; Ivezić et al. 2019) will only add to the data deluge. By collecting approximately 15 terabytes every night (Ivezić et al. 2019), the LSST will make the manual analysis and classification of data impossible.

The use of machine learning in astronomy has evolved rapidly over the last two decades (Baron 2019). While machine learning refers to any algorithm that learns a task by training on data, deep learning is a subset of machine learning in which representations are learned through multiple layers of neural networks. Development on this front has also benefited work in astronomy. Some examples of how machine learning and deep learning have aided astronomers are: star–galaxy classification (Philip et al. 2002; Ball et al. 2006; Vasconcellos et al. 2011; Abraham et al. 2012; Soumagnac et al. 2015; Kim & Brunner 2017; Clarke et al. 2020), stellar spectrum classification and interpolation (Kuntzer, Tewes & Courbin 2016; Sharma et al. 2020a, b), light-curve classification (Lochner et al. 2016; Mahabal et al. 2019; Möller & de Boissière 2020), galaxy morphology classification (Dieleman, Willett & Dambre 2015; Abraham et al. 2018; Domínguez Sánchez et al. 2018; Barchi et al. 2020; Walmsley et al. 2020; Nair & Vivek 2022), photometric redshift estimation (D'Isanto & Polsterer 2018; Pasquet et al. 2019), gravitational waves identification (George & Huerta 2018), and gravitational lensing identification (Cheng et al. 2020). Our aim in this paper is to use deep learning to distinguish between stars, quasars, and galaxies based on their images in the SDSS. In survey images, stars and quasars appear as point sources convolved with the point spread function (PSF), and the distinction between the two is conventionally made through their spectra, which requires large telescopes and is time-consuming even with multi-object spectrographs. Machine learning enables the separation of stars from quasars using their colours alone with good accuracy (see e.g. Abraham et al. 2012). A difficulty in this approach is that the colours of high-redshift quasars appear different from those of low-redshift quasars, which might intrinsically have the same spectrum. Galaxies can be distinguished from stars (and quasars) on the basis of their morphology, as in the early work by Sebok (1986), but the accuracy diminishes as the objects get fainter and more compact. A more reliable approach to classify in such cases is to use the difference in spectral energy distributions, either using spectra or using multiwavelength images. However, both these approaches are limited by the availability of large-scale data.

★ E-mail: atharvabagul2000@gmail.com

Table 1. List of photometric parameters used. The colours given in the rows 5–8 are based on the dereddened magnitude. Note: Here x denotes one of the filters *u*/*g*/*r*/*i*/*z*.

Name	Description	Total parameters
dered_x	Dereddened magnitude: corrected for extinction	5
deVRad_x	DeVaucouleurs fit scale radius	5
psffwhm_x	FWHM of the PSF	5
extinction_x	Extinction	5
u_g	Colour: $u - g$	1
g_r	Colour: $g - r$	1
r_i	Colour: $r - i$	1
i_z	Colour: $i - z$	1

As the availability of data has increased, machine learning has become a powerful tool for star–galaxy classification. Earlier methods used random forests (Vasconcellos et al. 2011) and support vector machines (Fadely, Hogg & Willman 2012), which are easier to train with relatively small amounts of data, but they have weak performance (80–90 per cent completeness). Newer methods involving convolutional neural networks (CNNs; Hao-ran, Ji-ming & Jun-yi 2017; Kim & Brunner 2017; Kennamer et al. 2018) perform significantly better, aided by the development of better image classification algorithms by the machine learning community for ImageNet 1 (Krizhevsky, Sutskever & Hinton 2012; Szegedy et al. 2014; He et al. 2015; Simonyan & Zisserman 2015).

There is often a faint limit for magnitudes beyond which classification usually becomes difficult. For example, current methods degrade in performance when $r > 22.5$ (Cabayol et al. 2019) and $i > 21$ (Kim & Brunner 2017). Newer sky surveys like the LSST will observe even deeper into the sky, making the importance of classification at faint magnitudes ever more important. We present a novel method to classify compact, faint sources with increased accuracy, in which we use a deep learning architecture that combines parametric and image-based classification. Signal-to-noise ratio plays an important role in running our model on these newer data sets. It consists of a mixture (stacking ensemble) of a CNN and an artificial neural network (ANN). Our training set consists of a set of photometric features and the images in five photometric bands (*u*, *g*, *r*, *i*, *z*) corresponding to the SDSS sources which have been spectroscopically classified as star, galaxy, or quasar by the SDSS pipeline. In order to select only the compact galaxies and fainter sources, we use some constraints on de Vaucouleur's radius and photometric magnitude which are defined and explained in Section 2. For the full application of `MargNet`, we need the measured parameters (as mentioned in the Table 1) for the machine learning part. For data sets from other surveys, the measured parameters may not be identical to those for the SDSS. In such cases, the machine learning network can be easily altered to accommodate the different set. If the parameters are not provided, then they could be measured by the user. If use of the parameters is not practical only the deep learning part could be used which needs just the images.[1] We perform three experiments to look at different aspects of performance in the faint and compact limit by preparing three different sets of training, validation, and test samples. Many past studies focus only on the star–galaxy separation.

Some earlier studies have also considered the full star galaxy and quasar separation problem (e.g. Xiao-qing & Jin-Meng 2020; Nakazono et al. 2021). To compare our results with those in recently published work, we separately perform star–galaxy classification as well as star–galaxy–quasar classification in each of the three experiments. We show that our method performs better than the earlier methods for faint and compact galaxies, considering the limitation on the faintness and compactness parameters and subsequent limits on them as defined in the Section 2.2.

This paper is structured as follows: Section 2 covers data acquisition from SDSS DR16 Catalogue. Section 3 covers the preprocessing and Section 4 the method employed for the classification problem along with details on the models. In Section 5, we provide the results and discussion and in Section 6 the conclusions and further possibilities that arise from this work.

2 DATA SET

2.1 SDSS data

Of the numerous sky surveys conducted over the past decades, few compare to the extent and detail of the SDSS (York et al. 2000). The SDSS collected data with a dedicated 2.5 m telescope (Gunn et al. 2006), CCD camera (Gunn et al. 1998), five filters (Doi et al. 2010), and a spectrograph (Smee et al. 2013). The SDSS furnishes detailed, multicoloured images of approximately one-third of the sky and spectra for more than three million astronomical objects. For this paper, we make use of the SDSS Data Release 16 (Ahumada et al. 2020).

We only use SDSS photometric data in the five passbands, namely *u*, *g*, *r*, *i*, and *z* (Fukugita et al. 1996) as an input to our models. Since we train our models in a supervised fashion, an approach which requires a label corresponding to each input, we obtain the spectroscopically assigned class by the SDSS pipeline.

We use two different types of data from the SDSS for our work: image data and photometrc features.

(i) The image data in the five photometric filters, *u*, *g*, *r*, *i*, *z*, is extracted from the FITS (Flexible Image Transport System) files obtained in each filter from the SDSS.

(ii) The photometric features[2] listed in Table 1 for each source are also used as input to the models. These features are obtained using the SDSS query service CasJobs.

We use two steps to control the quality of our data. First, we require that each object have an error in the PSF magnitude less than 0.2 across the *r* passband. Second, we use a combination of flags to limit our search to objects with clean photometry in our query. These flags[3] are:

[1] We have provided both pre-trained models on our GitHub repository mentioned in the data availability section.

[2] The definitions of all the parameters can be found at DR16 PhotObjAll.

[3] A detailed description of these flags is available on the SDSS website.

(i) s.zWarning = 0 (no problem with spectra)
(ii) p.clean = 1 (clean photometry)
(iii) p.mode = 1 (select primary objects only)
(iv) s.sciencePrimary = 1 (no repeats in this query)
(v) s.targetType = 'SCIENCE'
(vi) p.insideMask = 0 (photometry without any masks)
(vii) (p.flags_r & 0x20) = 0 (not PEAKCENTER)
(viii) (p.flags_r & 0x80000) = 0 (not NOTCHECKED)
(ix) (p.flags_r & 0x800000000000) = 0 (not PSF_FLUX_INTERP)
(x) (p.flags_r & 0x10000000000) = 0 (not BAD_COUNTS_ERROR)
(xi) ((p.flags_r & 0x100000000000) = 0 OR (p.flags_r & 0x1000) = 0) (not both INTERP_CENTER and COSMIC_RAY)
(xii) (p.flags_r & 0x40000) = 0 (not SATURATED)
(xiii) (p.flags_r & 0x80) = 0 (not NOPROFILE)

2.2 Compactness and faintness

As described in Section 1, we aim to identify galaxies which are too compact to properly separate from stars and quasars using traditional methods. Astronomers have used different ways of measuring the compactness of sources, which lead to nearly the same results. While some definitions of compactness are based on the galaxy's magnitude (Fairall 1978; Hickson 1982), we do not use any magnitude criterion in our work, because low surface brightness (LSB) galaxies can have extended morphology albeit being faint. Instead, we define the compactness parameter c for an object as

$$c = \left\langle \frac{\mathrm{deVRad}}{\mathrm{FWHM}} \right\rangle \tag{1}$$

Here, deVRad is the half-light radius (also known as de Vaucouleur's radius) of the galaxy. $\langle\rangle$ denotes an average over all the five passbands: u, g, r, i, and z. FWHM denotes the full width at half-maximum of the PSF. For a given FWHM, smaller the value of c, the more compact is the galaxy. To establish the limit on c below which classification becomes difficult, we trained a random forest classifier using photometric parameters (see Section 2 for details) on a small random data set of spectroscopically identified stars, galaxies, and quasars, and evaluated the performance using the accuracy achieved in the classification. By retraining the random forest for training samples in small bins of c, we can see how the value of c affects the performance, as shown in Fig. 1. We observe a sharp drop in the performance when c becomes <0.5. We therefore use $c = 0.5$ as the boundary below which we classify galaxies as compact. The drop in performance for values of $c > 0.5$ is due to the fewer number of galaxies available for training the random forest classifier.

This finding qualitatively implies that when c is less than or equal to 0.5 (the galaxy's diameter is roughly equal to the FWHM of its PSF), the automated classification becomes challenging due to the compactness of the galaxy.

When choosing objects to constitute our faint data set, we require that the average magnitude in the five passbands

$$\langle \mathrm{mag} \rangle > 20. \tag{2}$$

We choose 20 as the cut-off point for two reasons. First, it is beyond this level of faintness that the traditional star–galaxy classifiers start to fail, with a huge decrease in performance at $r > 21$ (Kim & Brunner 2017; Cabayol et al. 2019). Secondly, enough samples from the SDSS obey this cut-off. A neural network needs sufficient amount of data to train properly, and a higher cut-off would have resulted in too small a data set.

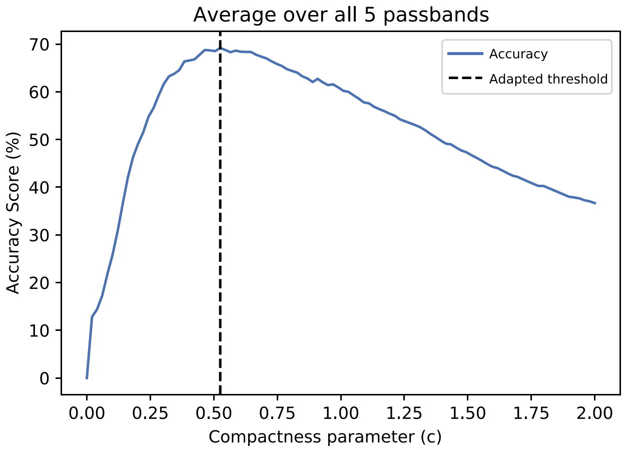


Figure 1. Model performance versus the compactness parameter, $c = \langle \frac{\mathrm{deVRad}}{\mathrm{FWHM}} \rangle$.

2.3 Querying faint and compact sources

As described in Section 2.2, we create two sets of data for our three experiments:

(i) Compact source data set
(ii) Faint and compact source data set

Because classification algorithms trained on an imbalanced data set perform poorly on minority classes, we prefer to work with a balanced number of classes throughout this paper.

2.3.1 *Compact source data set*

We impose the condition $c < 0.5$ from equation (1) for querying the marginal, compact galaxies. While the de Vaucouleur's radius is generally only applied to galaxies, we apply our condition to all samples here in order to maintain uniformity in our data.

In DR16, there are 173 813 stars, 80 779 galaxies, and 527 922 quasars which are spectroscopically identified and meet all of the above criteria for the compact source data set. We choose a random sample of 80 000 objects of each class to form our compact source data set.

2.3.2 *Faint and compact source data set*

In addition to the compact criterion $c < 0.5$ from equation (1), here we also require the condition $\langle mag \rangle > 20$ from equation (2). This gives us a data set which contains sources that are faint as well as compact.

In DR16, there are 55 323 stars, 74 503 galaxies, and 382 899 quasars that meet all of the above criteria and we choose a random sample of 50 000 objects of each class to form our faint and compact source data set. After querying the photometrc features and downloading the FITS images for both data sets, we are ready to run the three experiments.

3 PREPROCESSING

3.1 Image data

We use FITS images in the five passbands. However, there are a couple of issues that need to be solved during preprocessing. We

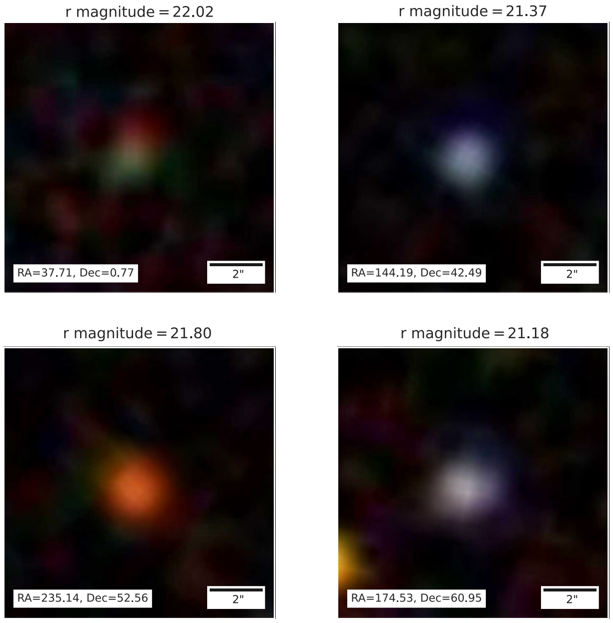


Figure 2. A sample of four compact galaxies from our data set. The images are constructed from the $g - r$, $r - i$, and $i - z$ colours obtained from the passbands.

crop each FITS image to a 32 × 32 pixel image to remove any undesired features or nearby sources from the input image. We find these dimensions sufficient for our compact objects of interest. The right ascension (RA) and declination (Dec.) values of the object of interest specify its pixel location in the image. This is used as a reference point for the image 'centre'. The image is cropped using the World Coordinate Systems on ASTROPY in PYTHON. We specify a 32 × 32 pixel bounding box aligned with the axes around the centre for this purpose. Since each passband has a separate cropped image, we stack them on top of each other. We thus produce a five-channel image (Fig. 2) of dimension 32 × 32 × 5 for each of the 240 000 objects.

3.2 Photometric features

The photometric parameters, as given in Table 1, have different ranges. Normalization of a data set is a common requirement for many machine learning algorithms. Thus, we normalize the data by removing the mean and scaling to unit variance for every individual parameter using `StandardScaler`[4] from the SCIKIT-LEARN (Pedregosa et al. 2011) library on PYTHON.

3.3 Train-validation-test split

As is common in deep learning, we split our data set into a training set, validation set, and a test set. In each of the three experiments, we select an approximately equal number of classes. Each subset is chosen differently depending on the experiment, full details of which are given in Table 2 and Fig. 3.

3.3.1 *Experiment 1*

In this experiment, all three sets – training, validation, and test, are chosen from the compact source data set (Section 2.3.1: $c < 0.5$), which is split in the ratio 6:1:1 (i.e. 75 per cent training, 12.5 per cent validation, and 12.5 per cent test). We note that the test set is representative of the training set here.

3.3.2 *Experiment 2*

In this experiment, all three sets – training, validation, and test, are chosen from the faint and compact source data set (Section 2.3.2: $c < 0.5$; $\langle mag \rangle > 20$) and split in the ratio 8:1:1 (i.e. 80 per cent training, 10 per cent validation, and 10 per cent test). Here, we split our data differently than Experiment 1 and Experiment 3 as there were just 50 000 objects from each class in our data set (as described in Section 2.3.2). So to ensure that our training set had enough training data, we changed our data split so that at least 40 000 objects of each class were used for training in each experiment. This resulted in us using the ratio 8:1:1 (i.e. 80 per cent training, 10 per cent validation and 10 per cent test). We note that the test set is representative of the training set here.

3.3.3 *Experiment 3*

In this experiment, the training and validation set is chosen from the compact source data set (Section 2.3.1: $c < 0.5$). However, the test set is chosen from the faint and compact source data set (Section 2.3.2: $c < 0.5$; $\langle mag \rangle > 20$), such that the ratio is 6:1:1 (i.e. 75 per cent training, 12.5 per cent validation, 12.5 per cent test). We also note that the test set is **not** representative of the training set here.

This experiment aims to see how well we can extend the training at brighter levels to fainter levels. This case will be significant when surveys like the LSST are in operation. This experiment aims to see how well we can extend the training at brighter levels to fainter levels. This case will be significant when surveys like the LSST are in operation. The spectroscopic data set containing both faint and compact sources will have a relatively small number of objects and will pose a challenge for training a photometric classifier. Instead, we wish to look at how a model trained only on compact sources tests on faint and compact sources.

4 METHODOLOGY

The approach consists of two parallel but separate machine learning models pipelined to create a functioning classifier. The first unit is a CNN (LeCun, Bengio & Hinton 2015) model that takes the preprocessed images (processed as per Sec 3) as the input. The second unit is an ANN model (McCulloch & Pitts 1943), which takes the photometric parameters as input. These two units individually make an initial classification. The ensemble takes the initial classifications of CNN and ANN units as inputs and makes the final star, galaxy, and quasar classification.

4.1 ANN architecture

ANNs draw inspiration from biological neural networks consisting of biological neurons as basic units. A biological neuron receives input signals through dendrites, processes the received information, and provides the output. Similarly, an ANN consists of artificial neurons where each neuron receives the input through input nodes, processes

[4] The documentation of StandardScaler function from SCIKIT-LEARN can be found here.

Table 2. The distribution of our data set for each experiment. We use an approximately equal of number of classes in each subset. In Experiment 1, all the sets consist of compact sources ($c < 0.5$), while in Experiment 2, all the sets consist of faint and compact sources ($c < 0.5$; $\langle mag \rangle > 20$). But, in Experiment 3, the training and validation set consist of compact sources ($c < 0.5$), while the test set consists of faint and compact sources ($c < 0.5$; $\langle mag \rangle > 20$).

Experiment No.	Data set	Subset	# Stars	# Galaxies	# Quasars	Total
Experiment 1	Compact	Train	59 988	60 061	59 962	180 011
		Validation	10 033	9918	10 037	29 988
		Test	9979	10 020	10 001	30 000
Experiment 2	Faint and compact	Train	39 870	40 134	39 992	119 996
		Validation	5095	4956	4949	15 000
		Test	5031	4910	5059	15 000
Experiment 3	Compact	Train	59 988	60 061	59 962	180 011
		Validation	10 033	9918	10 037	29 988
	Faint and compact	Test	9511	9511	9510	28 532

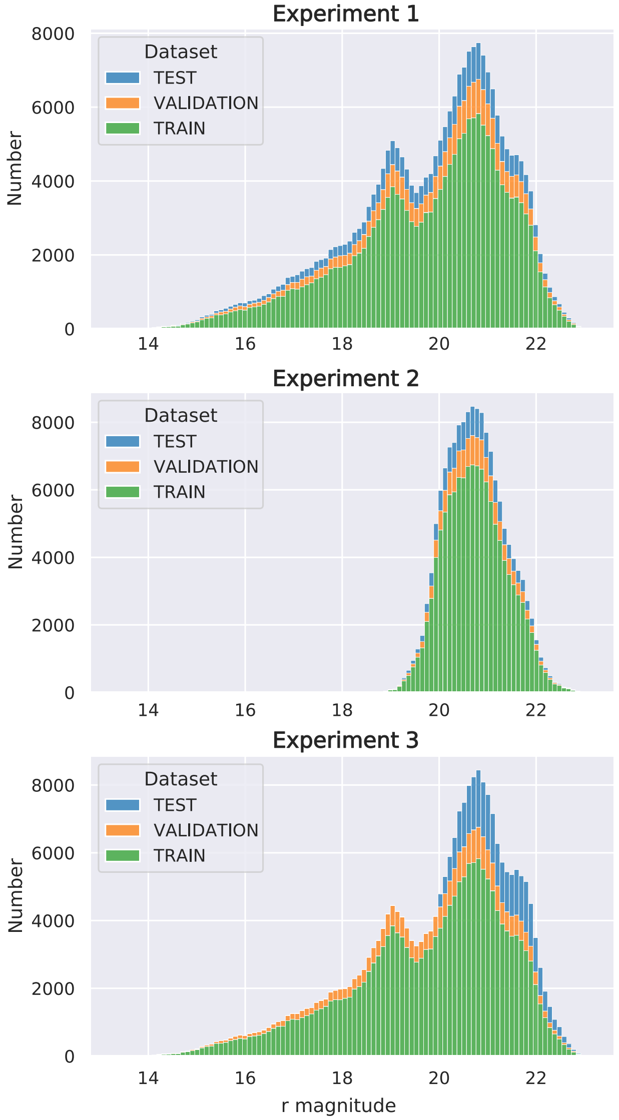


Figure 3. Number distribution of our data set as a function of magnitude in the r band for each experiment. Each image is a bar graph with each bar having train, validation, and test data in different colours being stacked above one another.

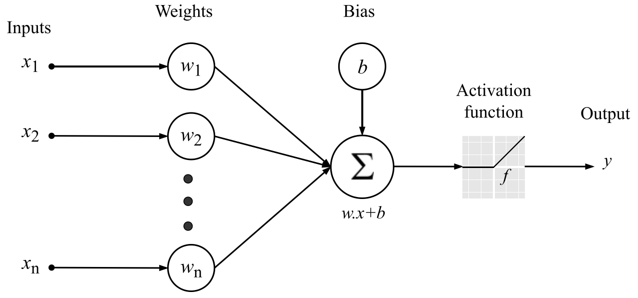


Figure 4. Schematic representation of an artificial neuron. Inputs (x_1, x_2,..., x_n) are multiplied by their corresponding weights (w_1, w_2,..., w_n). and a bias is added to the weighted inputs. An activation function is applied to the resultant which provides the final output y.

the information, and provides an output. Mathematically, the output y of an artificial neuron can be represented as follows:

$$y = f\left(\sum w \cdot x + b\right) \tag{3}$$

where x is a vector of inputs, (x_1, x_2,..., x_n), and w is a vector of weights (w_1, w_2,..., w_n) corresponding to each input. A bias value b is added to the summed input $w \cdot x$ which is further acted upon by a function f, referred to as the activation function. A schematic representation of an artificial neuron is shown in Fig. 4. A practical ANN consists of more than one such neuron arranged in multiple layers having interconnections between neurons in different layers. The training of the neural network refers to the learning of the weights w and bias value(s) b in order to minimize the departure between the neural output y and the expected output, say $\bar{y}$. A widely used non-linear activation function is the rectified linear unit (ReLU; Nair & Hinton 2010). Since ReLU units (equation 4), in general, allow faster training of deep neural networks with many layers, it was our natural choice for activation function in this exercise.

$$\sigma(x) = \begin{cases} x & x \geq 0, \\ 0.01x & x < 0 \end{cases} \tag{4}$$

4.2 CNN architecture

A CNN is a type of feed-forward neural network. These networks have gained significant use in computer vision applications like classification, counting, and a plethora of others. The main difference between a regular neural network (ANN) and a CNN is that the latter

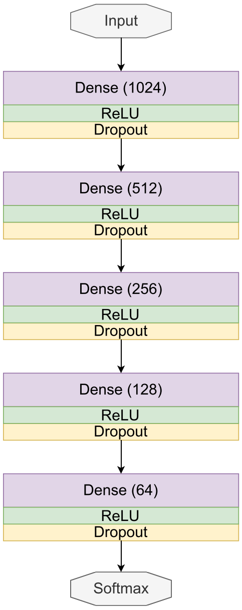


Figure 5. ANN architecture: This contains five dense layers of size 1024, 512, 256, 128, and 64, respectively, each activated with ReLU, along with a dropout fraction of 0.25. It trains a total of 331 331 neurons. For more details refer to Section 4.3.

has convolution operations between a set of filters and the inputs. A convolution is expressed as

$$y^k = \sigma\left(\sum_m w_m^k * x_m + b^k\right) \tag{5}$$

where we sum over the set of input feature maps, '$*$' is the convolution operator, and w represents the filter weights. Here, a feature map is the array of output activation obtained after applying the activation function. In a typical CNN, there are three kinds of layers: convolution layers, pooling layers, and fully connected layers. A pooling layer is present after a convolutional layer; it is used to to perform down-sampling of the feature map by way of a 'pooling' operation. This could, for example, lead to the average value of $n \times n$ pixel values, where n is chosen typically to be a small value like 2 or 3. This is done for all patches in the image, with stride length n and is carried forward as the output of an average-pooling layer, so that the image is down-sampled by a factor of n in each dimension. The primary function of the pooling layers is to reduce the dimension of the feature map and make the model invariant to small shifts and distortions. Similarly, max-pooling involves calculating the maximum value in a patch and using that as the layer's output. The fully connected layers are used at the end of a CNN which take the output of the convolutional and pooling layers to predict the best label that describes the image. The layers in CNNs are sparser than fully connected layers. In sparse connections, a neuron in one layer receives input from a local patch of neurons in the previous layer, whereas in fully connected layers the input received by a neuron is from every possible neuron in the previous layer. This sparsity reduces the number of connections in the CNN. Also, the kernel that is used for the convolution of a layer uses the same weights across the whole layer. This weight sharing greatly reduces the number of unique parameters in the network making CNNs relatively easier to train than fully connected networks.

A CNN takes in the pre-processed image files for all three data objects to provide an initial classification as the output. We have 80 000 images for each class with the available spectroscopic labels. The CNN uses only the images, without parameters. Each image is processed through a deep network of convolutional layers, of varying kernel sizes (kernels used: [1,1], [3,3], [5,5]), each activated by a ReLU unit (Nair & Hinton 2010). A dense layer is one where each neuron in the layer receives an input from every single neuron in the previous layer. Every possible pair has a weighted connection between its members. The information is distilled through such final few dense layers, and the final layer contains a softmax (generalization of the logistic regression function used for multi-class classification) function to divide each input as a probability distribution across the three classes. We employ a frequently used a widely used loss function, categorical cross-entropy. The CNN model trains a total of 25 544 807 neurons. We have used a CNN architecture which is inspired by the Inception Network created by Szegedy et al. (2014) for the ImageNet Challenge. InceptionNet has been used successfully in astronomy, and we employ the layer sizes used by Pasquet et al. (2019) for photometric redshift estimation. The inception modules are useful because they help extract the most discriminating features from different sizes of convolution filters. So filters of size (1 × 1), (3 × 3), and (5 × 5) are applied at one layer of the network, on whatever the input is at that depth. Each output is concatenated to form one thick layer. We perform this operation every few layers, and each such operation defines our 'inception module'. It is up to the network to weigh different filter outputs during training. The implementation details for this are available in Table 3.

We do not use dropout fraction (details for dropout fraction are given in Section 4.4) for CNN as it is used for densely connected layers, not in layers with sparse connections like the convolutional layers in the CNN. Dropout fraction is applied in neural networks which have significantly fully connected components (e.g. ANN). This prevents the large number of parameters present from overfitting the data.

4.3 Artificial neural network

In this component, the photometric parameters of the images are used for input. The images are not considered, and hence no additional preprocessing (other than normalization) is required for this component. It is a feed-forward network with stacked dense layers, each activated by a ReLU unit (Nair & Hinton 2010). The five dense layers ensure enough depth for direct classification from the selected list of parameters. The presence of dropouts (refer to the Section 4.4) (dropout fraction = 0.25) ensures overfitting is taken care of, which is a problem, especially in deep networks (see next section). The final layer is a softmax-activated with three outputs, one for each category. Again, categorical cross-entropy is a natural choice for the loss function. The model trains a total of 331 331 neurons and the implementation details are available in Table 4. Tables 3 and 4 (Figure 5) combined simultaneously can be used to implement `MargNet`, as given in Fig. 6.

4.4 Dropout and image augmentation

When neural networks are trained on small or homogeneous data sets, the models tend to overfit the data (Srivastava et al. 2014a). They do not generalize effectively to perform well on the validation and test data sets. To mitigate overfitting, a technique we employ is the inclusion of dropout in the network (Srivastava et al.). A fraction (called the dropout fraction) of neurons are randomly chosen and

Table 3. Characteristics of each layer of the CNN architecture: Column 1: name of the layer, Column 2: input layer, Column 3: number of filters in the convolution kernel, Column 4: size of the convolution kernel (in pixels), and Column 5: activation function used for the layer.

Layer	Input	# Filters	Kernel size	Activation
Conv2D_1	Input image	64	(5,5)	ReLU
Conv2D_2	Conv2D_1	48	(1,1)	ReLU
Conv2D_3	Conv2D_1	48	(1,1)	ReLU
Conv2D_4	Conv2D_1	48	(1,1)	ReLU
AvgPool2D_1	Conv2D_4	–	–	–
Conv2D_5	Conv2D_2	64	(1,1)	ReLU
Conv2D_6	Conv2D_2	64	(3,3)	ReLU
Conv2D_7	Conv2D_3	64	(5,5)	ReLU
Concatenate_1	[Conv2D_5, Conv2D_6. Conv2D_7, AvgPool2D_1]	–	–	–
Conv2D_8	Concatenate_1	64	(1,1)	ReLU
Conv2D_9	Concatenate_1	64	(1,1)	ReLU
Conv2D_10	Concatenate_1	64	(1,1)	ReLU
AvgPool2D_2	Conv2D_10	–	–	–
Conv2D_11	Conv2D_8	92	(1,1)	ReLU
Conv2D_12	Conv2D_8	92	(3,3)	ReLU
Conv2D_13	Conv2D_9	92	(5,5)	ReLU
Concatenate_2	[Conv2D_11, Conv2D_12, Conv2D_13, AvgPool2D_2]	–	–	–
AvgPool2D_3	Concatenate_2	–	–	–
Conv2D_14	AvgPool2D_3	92	(1,1)	ReLU
Conv2D_15	AvgPool2D_3	92	(1,1)	ReLU
Conv2D_16	AvgPool2D_3	92	(1,1)	ReLU
AvgPool2D_4	Conv2D_16	–	–	–
Conv2D_17	Conv2D_14	128	(1,1)	ReLU
Conv2D_18	Conv2D_14	128	(3,3)	ReLU
Conv2D_19	Conv2D_15	128	(5,5)	ReLU
Concatenate_3	[Conv2D_17, Conv2D_18, Conv2D_19, AvgPool2D_4]	–	–	–
Conv2D_20	Concatenate_3	92	(1,1)	ReLU
Conv2D_21	Concatenate_3	92	(1,1)	ReLU
Conv2D_22	Concatenate_3	92	(1,1)	ReLU
AvgPool2D_5	Conv2D_22	–	–	–
Conv2D_23	Conv2D_20	128	(1,1)	ReLU
Conv2D_24	Conv2D_20	128	(3,3)	ReLU
Conv2D_25	Conv2D_21	128	(5,5)	ReLU
Concatenate_4	[Conv2D_23, Conv2D_24, Conv2D_25, AvgPool2D_5]	–	–	–
AvgPool2D_6	Concatenate_4	–	–	–
Conv2D_26	AvgPool2D_6	92	(1,1)	ReLU
Conv2D_27	AvgPool2D_6	92	(1,1)	ReLU
AvgPool2D_7	Conv2D_27	–	–	–
Conv2D_28	AvgPool2D_6	128	(1,1)	ReLU
Conv2D_29	Conv2D_26	128	(3,3)	ReLU
Concatenate_5	[Conv2D_29, AvgPool2D_7, Conv2D_28]	–	–	–
Dense_1	Concatenate_5	1024*		None
Dense_2	Dense_1	1024*	–	None
Dense_3	Dense_2	3*	–	Softmax

Table 4. Characteristics of each layer of the ANN architecture: Column 1: name of the layer, Column 2: input layer, Column 3: refers to the number of neurons in the layer, Column 4: activation function used for the layer, and Column 5: value of the dropout function for each layer in the ANN.

Layer	Input	Neurons	Activation	Dropout
Dense_4	Photometric features	1024	ReLU	0.25
Dense_5	Dense_4	256	ReLU	0.25
Dense_6	Dense_5	128	–	0.25
Dense_7	Dense_6	32	ReLU	0.25
Dense_8	Dense_7	64	ReLU	0.25
Dense_9	Dense_8	3	Softmax	0

artificially set to 0 only during training. This helps prevent overfitting of the neural network.

Data augmentation is another effective method that can be used to minimize overfitting. We create variations in the data, which are fed to the model while training. This artificially created and enhanced version of the data improves model learning, enabling the model to generalize well during the testing phase (Perez & Wang 2017). We augment the procured data points using PYTHON KERAS tools, namely the `ImageDataGenerator` module[5] here, to make our model more robust. We apply the following transforms independently and randomly over the data set: (1) We mirror images randomly

[5] Details and documentation of the `ImageDataGenerator` module can be found.

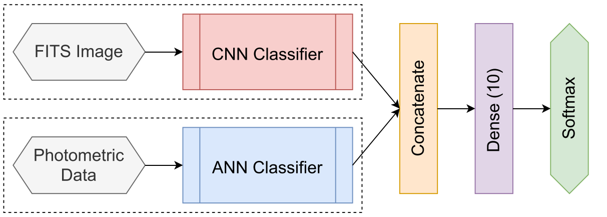


Figure 6. A schematic representation of MargNet. Details of the full implementation are given in the Tables 3 and 4. Note that the outputs of the CNN classifier and the ANN classifier consist of three neurons each, which represent the separate predicted probabilities. These are then 'concatenated' together and fed forward to a fully connected dense layer of 10 neurons. These are then finally fed to three softmax activated neurons, which represent MargNet's final probability predictions.

along the horizontal or the vertical axis; (2) We rotate images by an angle between 1° and 180°. The image background is mostly monochromatic and of little use; hence the cropping of edges during rotation is neglected; and (3) We translate images along the vertical and horizontal axis by 1 to 3 pixels. The resultant images are affine transformations of the originals, defined by

$$y = Wx + b. \tag{6}$$

Here, W is the *affine transformation matrix* used, y is the affine transformation map, and b is a bias vector. (Gallini Gallini). Affine transforms maintain parallelism and lines, and hence do not destroy or fabricate astronomical information in the images relevant to us.

4.5 Ensemble (MargNet)

The CNN and ANN were trained with an early stopping condition that the training will stop if it shows no improvement in 30 consecutive epochs. These previously trained models are both stacked parallel to each other, in a stacking ensemble. We call our final network MargNet. We have optimized the CNN and ANN for their respective inputs and hence do not retrain MargNet. We take the objects in each class (both the images and photometric data) and give them as inputs for the two models, CNN and ANN, respectively. The outputs of these models then act as inputs for the ensemble (a schematic representation is shown in Fig. 6). The expected outputs for both components are merged, and a statistical combination of them leads to the final output. We calculate the performance metrics for the final output. MargNet contains 103 neurons of its own and a total of 25 875 217 inclusive of CNN and ANN model trained neurons.

4.6 Evaluation metrics

To measure classification performance, we evaluate three metrics – precision, recall, and accuracy. Precision is the ratio of the true positives (TP) to the sum of true positives and false positives (FP), while recall (R) is the ratio of true positives to the sum of true positives and false negatives (FN).

Thus, for a particular class c, the precision (P_c) and the recall (R_c) are given by

$$P_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}, \quad R_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}, \tag{7}$$

where the subscript c denotes objects for that particular class only. We then calculate the total precision (P) and the total recall (R) by taking an average of the P_c and R_c across all classes.[6]

Intuitively, the precision for galaxies can be thought of as the classifier's ability to not misclassify a star/quasar as a galaxy, while the recall for galaxies denotes the classifier's ability to identify all galaxies in the data set.

Accuracy, on the other hand, is the number of correctly classified samples divided by the total number of classified samples. This can be calculated for just a single class, or for all classes. Accuracy is thus given by

$$\text{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{8}$$

where TN denotes the true negatives.

In this paper, all metrics refer to the total score across all classes, unless specified explicitly.

5 RESULTS AND DISCUSSION

For each experiment, we work with two cases: star–galaxy classification and star–galaxy–quasar classification. We use the ensemble model, MargNet.

5.1 Results: experiment 1

In this experiment, all three sets – training, validation, and test, are chosen from the compact source data set. We calculate the performance metrics for star–galaxy and star–galaxy–quasar classification. MargNet performs better than the ANN and CNN individually for both the classifications (see Table 5). For star–galaxy classification, we do not consider quasars because evaluating the performance for only stars and galaxies provides comparison with models used in other works. MargNet obtains an overall accuracy of 98.1 ± 0.1 per cent for the binary classification. We can also see from Fig. 7 a that performance on both stars and galaxies is similar (98.23 per cent for galaxies and 98.03 per cent for stars), with a similar number of misclassifications for both (1.77 per cent for galaxies and 1.97 per cent for stars).

For star–galaxy–quasar classification, MargNet obtains an average accuracy across the three classes of 93.3 ± 0.2 per cent. From the confusion matrices of Fig. 7(b), it is evident that performance for stars, galaxies, and quasars is different. Quasars have the lowest accuracy (91.04 per cent), and are misclassified as stars in ∼5.8 per cent of the cases, and as galaxies in ∼3.2 per cent of the cases. The more frequent identification of quasars as stars is understandable as both are point sources. For the same reason, stars are misclassified as quasars in ∼5.4 per cent of the cases. Galaxies have the best individual accuracy among the three classes. Galaxy-quasar misclassifications (galaxy–quasar: 2.92 per cent and quasar–galaxy: 3.15 per cent being the individual misclassification rates) are more common than galaxy–star misclassifications (galaxy–star: 1.20 per cent and star–galaxy: 1.51 per cent). This is due to quasar host galaxy fuzz, which can be apparent for low-luminosity quasars, leading to their being misclassifed as a compact galaxy in some cases. Stars have no structure other than the PSF and are therefore less likely to be classified as a galaxy.

[6] This is available on SCIKIT-LEARN as the macro setting.

Table 5. MargNet performance metrics: This table shows the performance of our models ANN, CNN, and the ensemble (MargNet) considering our compactness criterion mentioned in the Section 2.2, and faintness criteria based on the r-band magnitude. We show the performance metrics for all the three experiments which were carried out. Details of these experiments are mentioned in Section 1. We see that MargNet outperforms the individual models and gives good accuracy, and balanced precision and recall values for the star-galaxy and the star–galaxy–quasar problem for all the three experiments. The detailed confusion matrices for MargNet for both the problems can be seen in Fig. 7.

Experiment	Classification	Network	Accuracy (per cent)	Precision (per cent)	Recall (per cent)
Experiment 1	1. Star–galaxy	ANN	97.9 ± 0.1	97.8 ± 0.1	97.9 ± 0.1
		CNN	97.4 ± 0.1	97.4 ± 0.1	97.4 ± 0.1
		MargNet	98.1 ± 0.1	98.1 ± 0.1	98.1 ± 0.1
	2. Star–galaxy–quasar	ANN	93.0 ± 0.1	93.0 ± 0.1	93.0 ± 0.1
		CNN	91.6 ± 0.2	91.7 ± 0.2	91.6 ± 0.2
		MargNet	93.3 ± 0.2	93.3 ± 0.2	93.3 ± 0.2
Experiment 2	1. Star–galaxy	ANN	96.0 ± 0.1	96.0 ± 0.1	96.0 ± 0.1
		CNN	95.2 ± 0.1	95.2 ± 0.1	95.2 ± 0.1
		MargNet	96.9 ± 0.1	96.9 ± 0.1	96.9 ± 0.1
	2. Star–galaxy–quasar	ANN	86.0 ± 0.1	86.1 ± 0.1	86.1 ± 0.1
		CNN	84.2 ± 0.2	84.5 ± 0.2	84.2 ± 0.2
		MargNet	86.7 ± 0.2	86.8 ± 0.2	86.7 ± 0.2
Experiment 3	1. Star–galaxy	ANN	91.6 ± 0.1	92.4 ± 0.1	91.6 ± 0.1
		CNN	89.3 ± 0.1	90.4 ± 0.1	89.3 ± 0.1
		MargNet	92.0 ± 0.1	92.7 ± 0.1	92.0 ± 0.1
	2. Star–galaxy–quasar	ANN	73.1 ± 0.1	76.3 ± 0.1	73.1 ± 0.1
		CNN	69.4 ± 0.2	73.9 ± 0.2	69.4 ± 0.2
		MargNet	73.4 ± 0.2	76.5 ± 0.2	73.4 ± 0.2

5.2 Results: experiment 2

In this experiment, all three sets – training, validation, and test, are chosen from the faint and compact source data set. We evaluate the statistics as a function of the magnitude to study the performance of MargNet at fainter magnitudes (Table 5). For this, we divide the test set into bins of width 0.1 magnitude in the range $20 < r < 22.6$. Each bin in this range has at least 50 objects per class, which is a large enough sample size for the analysis. We evaluate different metrics for each bin and plot those in Fig. 8(a). We observe a decrease in performance as compared to Experiment 1 because the fainter an object is, the lower its is signal-to-noise ratio (SNR). This noise then propagates as an error in the object's photometric measurements and calculated features, making classification difficult.

For the star–galaxy binary classification problem, the trend we see (Fig. 8a) is that classification accuracy steadily decreases upto $r = 21.5$. We expect this, as the SNR in the photometric measurements reduces at fainter magnitudes. We note that there is an unexpected rise in accuracy for $r > 21.5$. We observe a similar trend for galaxy precision and star recall – a rise follows a drop. However, star precision and galaxy recall maintain a more or less constant value. This implies that, between $21 < r < 22$, the probability of a source predicted as a star being a star is very high. On the other hand, the probability of a source predicted as a galaxy not being a galaxy is very low.

In Fig. 8(a), for the tri-class classification problem of stars–galaxies–quasars, we see a trend similar to the one seen in Experiment 1 (Fig. 8). For classification accuracy, star recall, quasar recall, and quasar precision, a rise follows a drop, similar to the one we saw earlier in Section 5.2. Even the star precision and galaxy recall values are almost perfect and constant as earlier. However, the one that behaves differently is galaxy precision. At $r > 22$, unlike the star–galaxy case, no rise is seen. Instead, it just plateaus.

5.3 Results: experiment 3

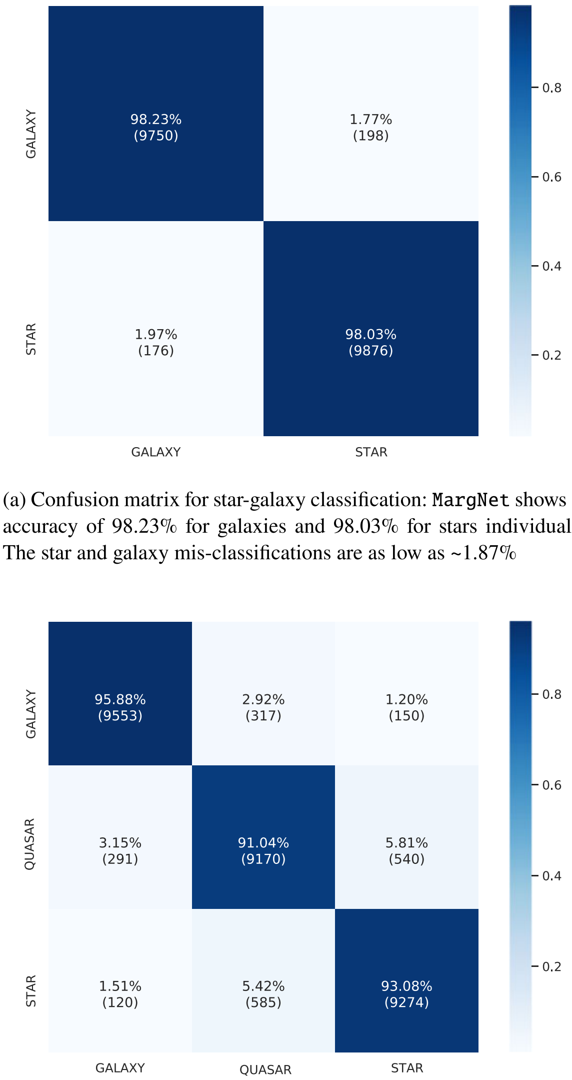
In this experiment, the training and validation set is chosen from the compact source data set, while the test set is chosen from the faint and compact source data set. Here, we present the performance at various magnitudes, as in Experiment 2, dividing the test set for Experiment 3 into bins of width 0.1 magnitudes. We plot the different metrics for each bin in Fig. 8(b). We expect a further decrease in performance when compared to Experiment 2, because the test set is not representative of the training set here. The training set consists of only compact sources, where the magnitude of the set covers a wide range from 14 to 23 (Fig. 3). However, the test set consists of only faint and compact sources in the range 20–22.6.

The trend we see from Fig. 8(b) is similar to Experiment 2 (Fig. 8a), albeit with a decreased performance overall, as expected. We see that even though the test set is not representative of the training set, it still performs fairly well for star–galaxy classification.

However, when considering quasars in the classification. The star–galaxy–quasar classification shows (Fig. 8b) that the performance reduces even before $r = 21$. Furthermore, the rise in performance at higher magnitudes is almost insignificant, which is unlike the star–galaxy case from Experiment 3 (Section 5.3) and Experiment 2 (Section 5.2).

5.4 Discussion

The paper by Sebok (1986) was an early foray into object classification and finding the magnitude limits for the confident separation of stars and galaxies. It aimed to find a magnitude to which a decision envelope can be drawn between stars and galaxies. This study for 130 000 objects found that objects fainter than $r = 19.6$ are not separated. The classification envelope crosses the decision boundary of $\phi = 1$. This criterion is a design of the classification and pattern recognition system used by the author, called a phi classifier,



(a) Confusion matrix for star-galaxy classification: `MargNet` shows an accuracy of 98.23% for galaxies and 98.03% for stars individually. The star and galaxy mis-classifications are as low as ~1.87%

(b) Confusion matrix for star-galaxy-quasar classification: `MargNet` shows an accuracy of 95.88% for galaxies, 91.04% for stars and 93.08% for quasars individually. The average mis-classification rate is ~3.33%

Figure 7. Confusion matrices of `MargNet` for Experiment 1: This shows the performance of `MargNet` for the both star–galaxy and the star–galaxy–quasar classification.

which works with linear data. `MargNet`, on the other hand, separates galaxies and stars with high completeness until $r = 21.5$ with a drop at fainter magnitudes. While other techniques after Sebok (1986) successfully classified faint objects with $r > 19.6$, we demonstrate that `MargNet` yields better performance than some well-known results as mentioned below.

In Fig. 8, we observe that the accuracy starts rising up beyond the 21.5 mag in the r band for both the star–galaxy and star–galaxy–quasar problems. We do not expect such a trend, because fainter galaxies are more difficult to classify, and should thus lead to a low accuracy. A possible reason for this could be that our training data set consists of an equal number of stars, galaxies, and quasars but their distribution for various r-band magnitudes is different – there are 2831 galaxies, 190 stars, and 660 quasars in our training set which satisfy the condition $r > 21.5$.

On the other hand, during testing, we use an equal number of all three objects across each magnitude bin, to ensure that the performance statistics for a particular bin are not influenced because of class imbalance. We believe that due to this large bias during training, the model's behaviour changes. Galaxies are less likely to be classified as another object, relatively reducing its number of false negatives. This effectively reduces the number of false positives for stars and quasars, and likely contributes to the increased precision.

To compare `MargNet`'s results with results in Kim & Brunner (2017), we plot the precision (purity) and recall (completeness)[7] again but impose additional conditions.[8] For galaxies, we plot the precision while maintaining constant recall ($c_g\ = 0.96$) (Figure9, Figure10). For stars, we plot the recall while maintaining constant precision ($p_g\ = 0.97$). We observe that `MargNet` consistently outperforms Kim & Brunner (2017) when comparing galaxies in the faint regime, $r > 21$, as per our aim. We also notice a sharp minimum between $20 < r < 20.25$ for galaxy precision, which is unexplained. For stars, `MargNet`'s recall exceeds Kim & Brunner (2017), after which it sharply drops at around $r = 21.5$. We would also like to clarify some gaps in both plots: we could not achieve the additional criterion ($c_g = 0.96$ for galaxies and $p_s = 0.97$ for stars) at these points.

Soumagnac et al. (2015) considered star–galaxy separation in the context of their work on Large Scale Structure and Weak Lensing using the DES. Comparing our results with them we find that `MargNet` performs better in the $r = 20$–20.5 regime. However, their model performs better for $r = 20.5$–23 mag as the survey that they use is a deep and wide survey, with a large number of objects at the faint magnitudes for the training.

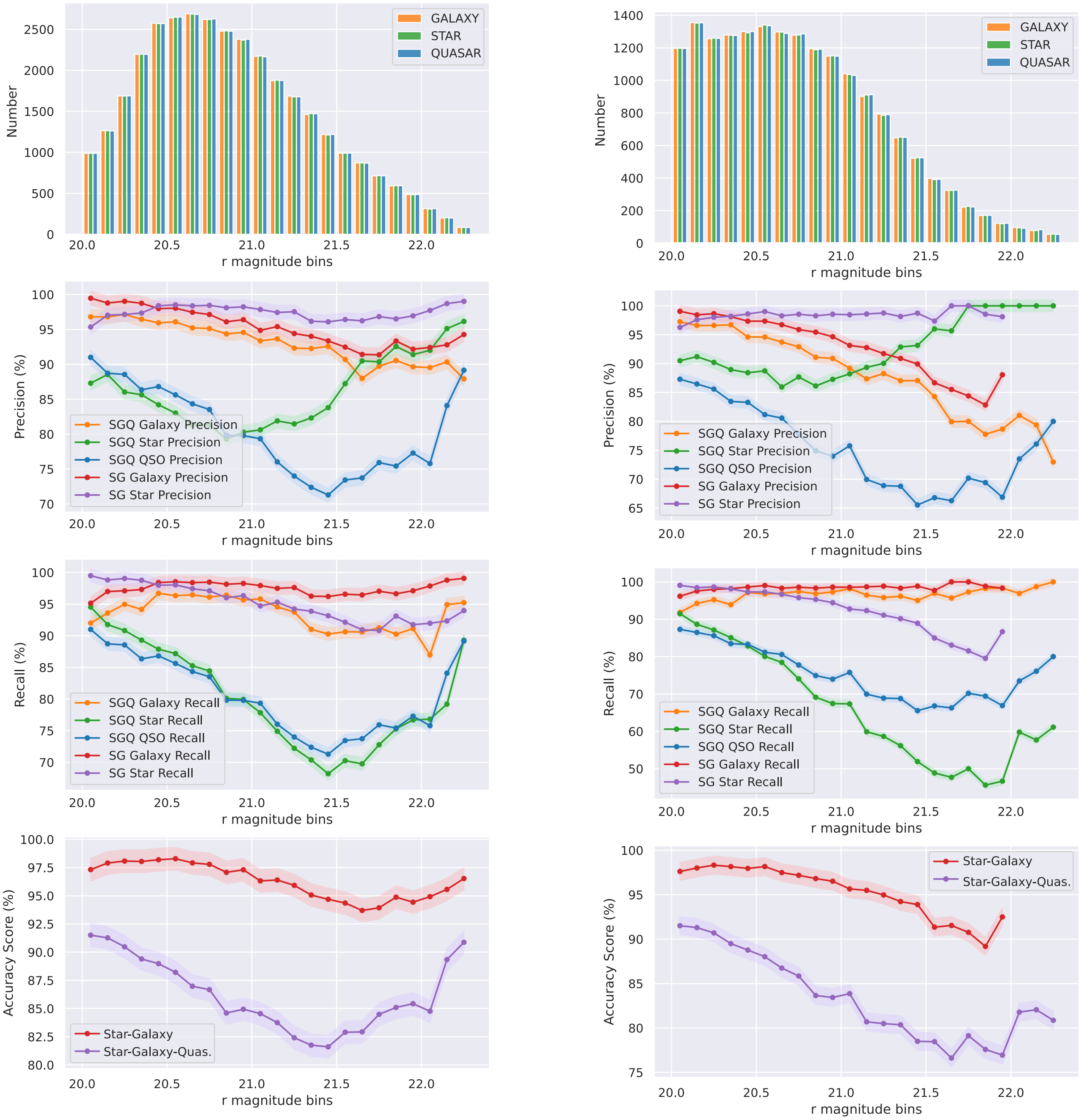
We can also compare our results with star–galaxy–quasar classification carried for other data sets (e.g. Xiao-qing & Jin-Meng 2020; Nakazono et al. 2021). We note that such comparison is not straightforward for data from different surveys with different data selection criteria. Xiao-qing & Jin-Meng (2020) used different machine learning approaches to classify stars, galaxies, and quasars from the Large Sky Area Multi-Object Fiber Spectroscopic telescope (LAMOST; Cui et al. 2012), and obtained their best accuracy (97.3 per cent) using a random forest classifier. Nakazono et al. (2021) compared the performance of Support Vector Machines and Random Forests for the classification of stars/galaxies/quasars in the Southern Photometric Local Universe Survey (S-PLUS; Mendes de Oliveira et al. 2019), supplemented by data from the *Wide-field Infrared Survey Explorer* (*WISE*; Wright et al. 2010). They achieved an averaged F-score of 96.85 ± 0.05 by using a random forest with 12 S-PLUS bands + 2 *WISE* bands.

6 CONCLUSION AND FUTURE SCOPE

We have presented a multiple neural network model named `MargNet` for classifying stars, galaxies, and quasars in the SDSS photometric images. For the SDSS data set, `MargNet` can provide a classification that is more accurate than previous works involving neural networks for compact and faint galaxies. The major advantage of `MargNet` is that useful features are learned automatically from

[7] Purity and completeness are conventional terms used in earlier astronomy literature, and mean the same as precision and recall, respectively.

[8] Here, we note that the training set of Kim & Brunner (2017) is different and the comparison is not easy as we cannot comment on whether the different results are due to the use of `MargNet` or due to different sky cuts for the data.



(a)

(b)

Figure 8. (a) Star–galaxy (SG) and star–galaxy–quasar (SGQ) classification for Experiment 2: For SG, we see accuracy sharply decreases at $r = 21$ and an unexpected rise for $r > 22$. For SGQ, similar trend as from SG case. In addition to SG, for quasar recall and quasar precision a rise follows a drop can be seen. At $r > 22$ there is no rise seen in galaxy precision. For further details refer to Section 5.2. (b) SG and SGQ classification for Experiment 3: For SG, the trends are similar to Experiment 2. For SGQ, the trends are similar to Experiment 2 (Fig. 8a). Here, for quasar recall and quasar precision a rise follows a drop can be seen. At $r > 22$ there is no rise seen in galaxy precision as compared to Fig. 8(a). Performance for Experiment 2: For the faint and compact data set $c < 0.5$; $r > 20$. The data are divided into bins of width 0.1 mag with the final range in $20 < r < 22.6$. Precision, recall, and accuracy metrics are evaluated for star–galaxy and star–galaxy–quasar problem, shown in Fig. 8(a). Performance for Experiment 3: Here, training is done on the compact data set $c < 0.5$, but testing is done on the faint and compact data set $c < 0.5$; $r > 20$. This data are divided into bins of 0.1 mag with the final range in $20 < r < 22.6$. Precision, recall, and accuracy metrics are evaluated for star–galaxy and star–galaxy–quasar problem, shown in Fig. 8(b).

images, while traditional machine learning algorithms require feature engineering as a separate process to produce accurate classifications. A part of MargNet is built by inception net modules. Inception Nets have recently achieved promising results in many image classification tasks and have been quickly and widely adopted by the computer vision community (Xiong, Yuan & Wang 2018). Usual networks show low accuracy when the sources become increasingly faint and compact, but MargNet shows promising results for such sources, with an overall accuracy of ∼ 93.3 per cent. For these faint and compact sources, the CNN and ANN individually perform with

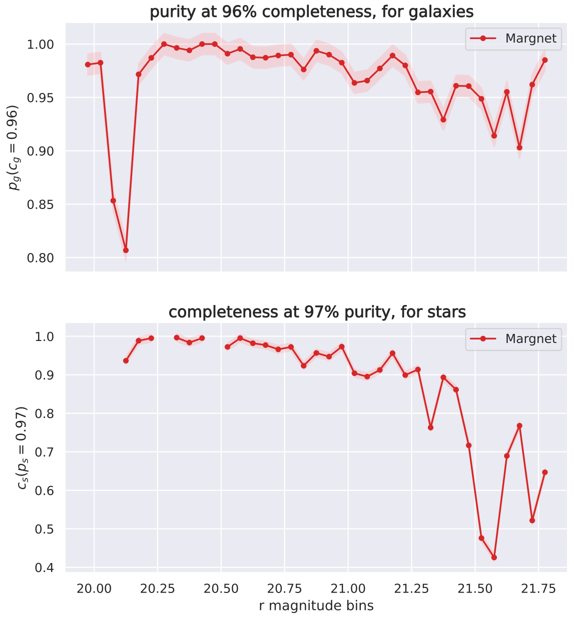


Figure 9. Galaxy purity (precision) and star completeness (recall) with additional conditions ($c_g = 0.96$; $p_s = 0.97$) as a function of the r-band magnitude. Note that we use i magnitude in the x-axis to compare it with fig. 8 from Kim & Brunner (2017).

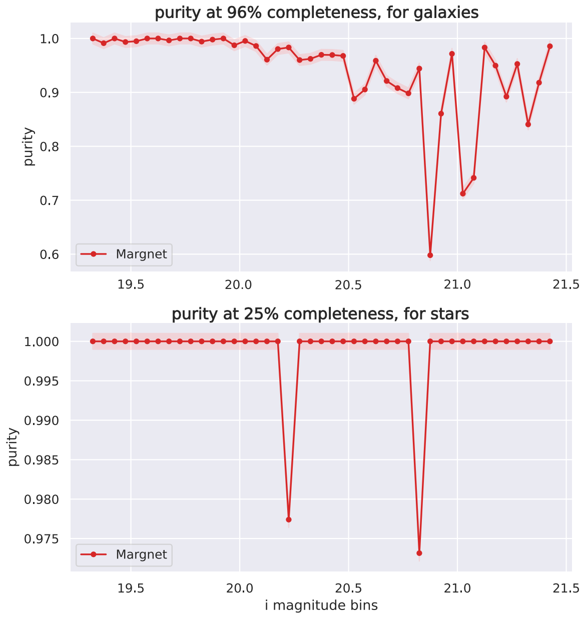


Figure 10. Galaxy purity (precision) and star purity with additional conditions on completness (recall) ($c_g = 0.96$; $c_s = 0.25$) as a function of the r-band magnitude. This plot can be compared with fig. 10 from Soumagnac et al. (2015).

~91.6 per cent and ~93.0 per cent accuracy, respectively. This denotes improvement in accuracy when both parts are run together. `MargNet` should be useful for future surveys like LSST which will capture many faint and compact sources. `MargNet` should also be useful for other surveys like GAIA and ZTF using cross-matching to SDSS and transfer learning. Although we will need photometric features to use the ensemble model, these features need not be identical to those from the SDSS. This could be done by altering the ANN network to accommodate other available features. Even in the case no such photometric features are obtainable, the image-based CNN component of `MargNet` could be used individually. The methodology behind `MargNet` will thus be of use with its accuracy in classifying objects as seen by such deeper astronomical surveys.

ACKNOWLEDGEMENTS

SC and AB acknowledge the computing resources provided by the Computer Centre, IISER Bhopal. MV acknowledges support from DST-SERB in the form of the core research grant (CRG/2020/1657). AB acknowledges support from Ms. Namita Yadav in the form of visualizations of models using figures.

We are grateful to the Sloan Digital Sky Survey for making their data free and open source. Funding for SDSS IV has been provided by the Alfred P. Sloan Foundation, the U.S. Department of Energy Office of Science, and the Participating Institutions. SDSS-IV acknowledges support and resources from the Center for High Performance Computing at the University of Utah. The SDSS website is www.sdss.org. SDSS-IV is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS Collaboration including the Brazilian Participation Group, the Carnegie Institution for Science, Carnegie Mellon University, Center for Astrophysics | Harvard & Smithsonian, the Chilean Participation Group, the French Participation Group, Instituto de Astrofísica de Canarias, The Johns Hopkins University, Kavli Institute for the Physics and Mathematics of the Universe (IPMU) / University of Tokyo, the Korean Participation Group, Lawrence Berkeley National Laboratory, Leibniz Institut für Astrophysik Potsdam (AIP), Max-Planck-Institut für Astronomie (MPIA Heidelberg), Max-Planck-Institut für Astrophysik (MPA Garching), Max-Planck-Institut für Extraterrestrische Physik (MPE), National Astronomical Observatories of China, New Mexico State University, New York University, University of Notre Dame, Observatário Nacional / MCTI, The Ohio State University, Pennsylvania State University, Shanghai Astronomical Observatory, United Kingdom Participation Group, Universidad Nacional Autónoma de México, University of Arizona, University of Colorado Boulder, University of Oxford, University of Portsmouth, University of Utah, University of Virginia, University of Washington, University of Wisconsin, Vanderbilt University, and Yale University.

Software: KERAS (Chollet et al. 2015), TENSORFLOW (Abadi et al. 2015), NUMPY (van der Walt, Colbert & Varoquaux 2011; Harris et al. 2020), JUPYTER (Kluyver et al. 2016), MATPLOTLIB (Hunter 2007), SEABORN (Waskom 2021), SCIKIT-LEARN (Pedregosa et al. 2011), SCIPY (Virtanen et al. 2020), PANDAS (Wes McKinney 2010; Reback et al. 2022), TQDM (da Costa-Luis et al. 2022), KERASTUNER (O'Malley et al. 2019), and PYTHON3 (Van Rossum & Drake 2009).

DATA AVAILABILITY

All our code used for this paper has been made open source on GitHub at the URL: https://github.com/sidchaini/MargNet. We downloaded table data from SDSS Casjobs DR16,[9] using the query listed in Appendix A. We downloaded image data from SDSS[10] using a PYTHON script which is available on our GitHub repository. Our

[9] https://skyserver.sdss.org/casjobs/

[10] https://data.sdss.org/sas/dr16/eboss/photoObj/

complete preprocessed data set and our trained models are available on Zenodo[11] (Chaini et al. 2022).

REFERENCES

Abadi M. et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems
Abraham S., Philip N. S., Kembhavi A., Wadadekar Y. G., Sinha R., 2012, MNRAS, 419, 80
Abraham S., Aniyan A. K., Kembhavi A. K., Philip N. S., Vaghmare K., 2018, MNRAS, 477, 894
Ahumada R. et al., 2020, ApJS, 249, 3
Ball N. M., Brunner R. J., Myers A. D., Tcheng D., 2006, ApJ, 650, 497
Barchi P. H. et al., 2020, Astron. Comput., 30, 100334
Baron D., 2019, preprint (arXiv:1904.07248)
Bellm E., 2014, preprint (arXiv:1410.8185)
Cabayol L. et al., 2019, MNRAS, 483, 529
Chaini S., Bagul A., Deshpande A., Gondkar R., Sharma K., Vivek M., Kembhavi A., 2022
Cheng T.-Y., Li N., Conselice C. J., Aragón-Salamanca A., Dye S., Metcalf R. B., 2020, MNRAS, 494, 3750
Chollet F. et al., 2015, Keras
Clarke A. O., Scaife A. M. M., Greenhalgh R., Griguta V., 2020, A&A, 639, A84
Cui X.-Q. et al., 2012, Res. Astron. Astrophys., 12, 1197
D'Isanto A., Polsterer K. L., 2018, A&A, 609, A111
da Costa-Luis C. et al., 2022
Dieleman S., Willett K. W., Dambre J., 2015, MNRAS, 450, 1441
Doi M. et al., 2010, AJ, 139, 1628
Domínguez Sánchez H., Huertas-Company M., Bernardi M., Tuccillo D., Fischer J. L., 2018, MNRAS, 476, 3661
Fadely R., Hogg D. W., Willman B., 2012, ApJ, 760, 15
Fairall A. P., 1978, The Observatory, 98, 1
Flaugher B., 2005, Int. J. Mod. Phys. A, 20, 3121
Fukugita M., Ichikawa T., Gunn J. E., Doi M., Shimasaku K., Schneider D. P., 1996, AJ, 111, 1748
George D., Huerta E. A., 2018, Phys. Lett. B, 778, 64
Gunn J. E. et al., 1998, AJ, 116, 3040
Gunn J. E. et al., 2006, AJ, 131, 2332
Hao-ran Q., Ji-ming L., Jun-yi W., 2017, Chin. Astron. Astrophys., 41, 282
Harris C. R. et al., 2020, Nature, 585, 357
He K., Zhang X., Ren S., Sun J., 2015, preprint (arXiv:1512.03385)
Hickson P., 1982, ApJ, 255, 382
Hunter J. D., 2007, Comput. Sci. Eng., 9, 90
Ivezić Ž. et al., 2019, ApJ, 873, 111
Kennamer N., Kirkby D., Ihler A., Sanchez-Lopez F. J., 2018, in Dy J., Krause A., eds, Proceedings of Machine Learning Research Vol. 80, Proceedings of the 35th International Conference on Machine Learning. PMLR, p. 2582
Kim E. J., Brunner R. J., 2017, MNRAS, 464, 4463
Kluyver T. et al., 2016, in Loizides F., Schmidt B., eds, Positioning and Power in Academic Publishing: Players, Agents and Agendas. p. 87
Krizhevsky A., Sutskever I., Hinton G. E., 2012, Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12. Curran Associates Inc., Red Hook, NY, p. 1097
Kuntzer T., Tewes M., Courbin F., 2016, A&A, 591, A54
LeCun Y., Bengio Y., Hinton G., 2015, Nature, 521, 436
Lochner M., McEwen J. D., Peiris H. V., Lahav O., Winter M. K., 2016, ApJS, 225, 31
McCulloch W. S., Pitts W., 1943, Bull. Math. Biophys., 5, 115
Mahabal A. et al., 2019, PASP, 131, 038002
Mendes de Oliveira C. et al., 2019, MNRAS, 489, 241
Miyazaki S. et al., 2002, PASJ, 54, 833
Möller A., de Boissière T., 2020, MNRAS, 491, 4277
Nair V., Hinton G. E., 2010, Proceedings of the 27th International Conference on International Conference on Machine Learning. ICML'10. Omnipress, Madison, WI, p. 807
Nair A., Vivek M., 2022, MNRAS, 511, 4946
Nakazono L. et al., 2021, MNRAS, 507, 5847
O'Malley T. et al., 2019, KerasTuner
Pasquet J., Bertin E., Treyer M., Arnouts S., Fouchez D., 2019, A&A, 621, A26
Pedregosa F. et al., 2011, J. Mach. Learn. Res., 12, 2825
Perez L., Wang J., 2017, preprint (arXiv:1712.04621)
Philip N. S., Wadadekar Y., Kembhavi A., Joseph K. B., 2002, A&A, 385, 1119
Reback J. et al., 2022, Pandas-Dev/Pandas: Pandas 1.4.2. Zenodo
Sebok W. L., 1986, ApJS, 62, 301
Sharma K., Kembhavi A., Kembhavi A., Sivarani T., Abraham S., Vaghmare K., 2020a, MNRAS, 491, 2280
Sharma K. et al., 2020b, MNRAS, 496, 5002
Simonyan K., Zisserman A., 2015, preprint (arXiv:1409.1556)
Smee S. A. et al., 2013, AJ, 146, 32
Soumagnac M. T. et al., 2015, MNRAS, 450, 666
Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014a, J. Mach. Learn. Res., 15, 1929
Szegedy C. et al., 2014, preprint (arXiv:1409.4842)
van der Walt S., Colbert S. C., Varoquaux G., 2011, Comput. Sci. Eng., 13, 22
Van Rossum G., Drake F. L., 2009, Python 3 Reference Manual. CreateSpace, Scotts Valley, CA
Vasconcellos E. C., de Carvalho R. R., Gal R. R., LaBarbera F. L., Capelato H. V., Frago Campos Velho H., Trevisan M., Ruiz R. S. R., 2011, AJ, 141, 189
Virtanen P. et al., 2020, Nat. Methods, 17, 261
Walmsley M. et al., 2020, MNRAS, 491, 1554
Waskom M. L., 2021, J. Open Source Softw., 6, 3021
Wes M., 2010, in van der Walt S., Millman J., eds, Proceedings of the 9th Python in Science Conference. p. 56
Wright E. L. et al., 2010, AJ, 140, 1868
Xiao-qing W., Jin-Meng Y., 2020, Chin. J. Phys., 69, 303
Xiong Z., Yuan Y., Wang Q., 2018, IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium. p. 2647
York D. G. et al., 2000, AJ, 120, 1579

APPENDIX A: SDSS SQL QUERY

We used the following SQL query on CasJobs to obtain all photometric features used in our work.

[11] https://doi.org/10.5281/zenodo.6659435

```
SELECT TOP 50000 p.objid, s.specobjid,
s.class,p.run, p.rerun, p.camcol, p.field,
p.ra, p.dec, s.z as redshift,
p.dered_u, p.deVRad_u, p.psffwhm_u, p.extinction_u,
p.dered_g, p.deVRad_g, p.psffwhm_g, p.extinction_g,
p.dered_r, p.deVRad_r, p.psffwhm_r, p.extinction_r,
p.dered_i, p.deVRad_i, p.psffwhm_i, p.extinction_i,
p.dered_z, p.deVRad_z, p.psffwhm_z, p.extinction_z,
p.u - p.g AS u_g,
p.g - p.r AS g_r,
p.r - p.i AS r_i,
p.i - p.z AS i_z
FROM SpecObj as s
JOIN PhotoObj AS p ON p.objid  =  s.bestObjID
WHERE s.z > 0 AND s.zErr<0.1
AND s.class  =  ’STAR’ -- Put STAR/GALAXY/QSO
AND s.bestObjID  =  s.fluxObjID
--For faint objects
AND (p.u+p.g+p.r+p.i + p.z)/5 > 20
AND p.psffwhm_u !=0 AND p.psffwhm_g ! = 0
AND p.psffwhm_r ! = 0
AND p.psffwhm_i !=0 AND p.psffwhm_z ! = 0
--For compact objects
AND ((p.deVRad_u / p.psffwhm_u)  +
   (p.deVRad_g / p.psffwhm_g)  +
   (p.deVRad_r / p.psffwhm_r)  +
   (p.deVRad_i / p.psffwhm_i)  +
   (p.deVRad_z / p.psffwhm_z)) / 5 < 0.5
AND ((p.err_u/p.u) + (p.err_g/p.g)  +
   (p.err_r/p.r) + (p.err_i/p.i)  +
   (p.err_z/p.z)) / 5 < 0.1
AND p.deVRad_u != 0 AND p.deVRad_g ! = 0
AND p.deVRad_r ! = 0
AND p.deVRad_i !=0 AND p.deVRad_z ! = 0
-- Quality flags
AND s.zWarning=0 AND p.clean=1 AND p.mode  =  1
AND s.sciencePrimary  =  1
AND s.targetType  =  ’SCIENCE’
AND p.insideMask = 0
AND (p.flags_r & 0x20)  =  0
AND (p.flags_r & 0x80000)  =  0
AND (p.flags_r & 0x800000000000)  =  0
AND (p.flags_r & 0x10000000000)  =  0
AND ((p.flags_r & 0x100000000000)  =  0
OR (p.flags_r & 0x1000)  =  0)
AND (p.flags_r & 0x40000)  =  0
AND (p.flags_r & 0x80)  =  0
AND (p.flags_r & 0x4)  =  0
AND ((p.flags_r & 0x10000000) ! =  0)
AND ((p.flags_r & 0x8100000c00a0)  =  0)
AND (((p.flags_r & 0x400000000000)  =  0)
AND (p.psfmagerr_r < =  0.2))
AND (((p.flags_r & 0x100000000000)  =  0)
AND (p.flags_r & 0x1000)  =  0);
```

This paper has been typeset from a TeX/LaTeX file prepared by the author.