Galaxy Classification using CNNs

Bellavia Gabriele, Belli Luigi, Lera Margherita, Maida Giovanni Andrea

Università di Padova

Submitted June 5, 2025

ABSTRACT

Context. The advancement in Deep Learning tools along with the rise in computational capabilities has unlocked new ways of efficiently analysing the vast amount of data scientific research produces. A supervised learning method trained on a dataset such as Galaxy Zoo 2 should be able to handle this income, and to greatly mitigate the workload on human researchers.
Aims. Be able to classify images of galaxies taken from the Galaxy Zoo 2 dataset.
Methods. Using an automated approach to subdivide the images into morphology categories. We tested the prowess of Convolutional Neural Networks in classifying galaxies, trying multiple architectures and selecting the most adequate.
Results. The implemented architectures proved themselves able to proficiently distinguish the major morphology classes such as ellipticals and spirals, and also to distinguish early and late-type morphologies for each class.

Key words. CNNs – GalaxyZoo2 – Image Classification – Machine Learning – Galaxy Morphology

1. Introduction

Since the first classification proposed by Hubble (1936), galaxy morphology is still a baseline for understanding galaxies. Morphology is strongly correlated with galactic star formation history, i.e. galaxies where star formation ceased gigayears ago are usually different from those where star formation continues to the present day Buta (2011). Deep surveys have extended this study beyond nearby galaxies, making it almost impossible to compile catalogues of classified galaxies by individuals or small teams of astronomers. A new method was proposed by Galaxy Zoo (Lintott et al. 2008), a web-based project which combined the classifications of images of galaxies drawn from the SDSS from more than 100 000 public participants. This project was able to produce a catalogue of more than 300 000 galaxies in agreement with those compiled by professional astronomers to an accuracy of better than 10%. After its success, Galaxy Zoo 2 (Willett et al. 2013) was launched, followed by many other crowdsourced projects. [1]

2. The Dataset

A subset of the Galaxy Zoo 2 image set was used in 2013 for a public challenge–hosted on the Kaggle platform–called *Galaxy Zoo: The Galaxy Challenge* [2]. The competition was aimed at finding an automated method able to reproduce the classification labels of all the images within the dataset.

The Galaxy Zoo Challenge dataset is composed by 61 578 labeled galaxy images, plus 79 978 unlabeled images. The images are 424×424 pixels in size and colour composite, stored in `jpeg` format. The queried object is at the center of each image. The images' labels are the summary of the public's answers to the proposed classification questions.

The decision tree that every user follows in the classification process is shown in Figure 1. Each question (or node) is represented by a set of labels, which are defined as the fraction of people who gave the specific answer to a certain question. They are related to the probabilities for each category, meaning that a high number (close to 1) indicates that many users identified this morphology category for the galaxy with a high level of confidence.

The final set of labels for each image is composed of 37 probability values. Each of these are multiplied by the probability of the previous choice in the decision tree, with the exception of the question *'Is there anything odd?'*, which is re-normalized to one.

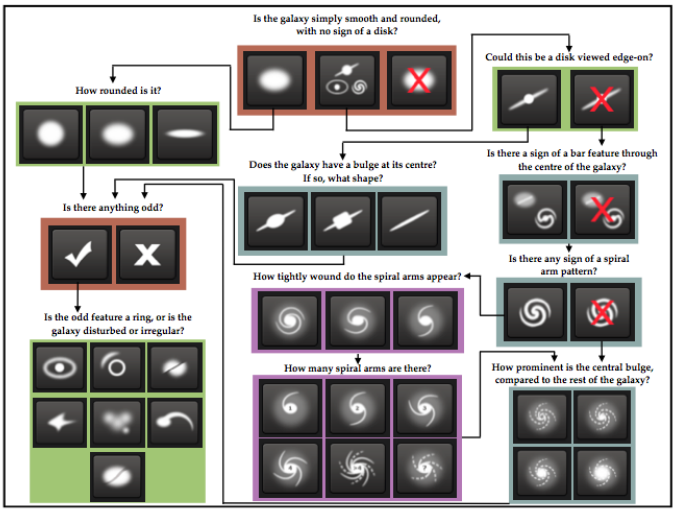


Fig. 1: Flowchart of the classification tasks for Galaxy Zoo 2, beginning at the top centre.

[1] `https://www.zooniverse.org/projects/zookeeper/galaxy-zoo`
[2] `https://www.kaggle.com/competitions/galaxy-zoo-the-galaxy-challenge/`

3. Methodology

Although the Galaxy Zoo projects were able to scale up the dimension of the classified galaxy catalogues in a short period of time via crowdsourcing, this approach does not scale well enough to keep up with the increasing availability of galaxy images that can be classified (Dieleman et al. 2015). With the launch of new telescopes, such as JWST and Euclid, and the continuous increase of data from various surveys, an automated approach is becoming indispensable. Thanks to the Galaxy Zoo projects, large training sets of reliably annotated images are available, allowing the training of more and more sophisticated machine learning models for automated morphological classification (Willett et al. 2013). This approach greatly reduces the experts' workload without affecting accuracy.

Among many algorithms, Deep Neural Networks in particular tend to scale very well as the number of available training examples increases. For their speed and accuracy, they have become the state-of-the-art approach to estimate galaxy morphology in large datasets. **Convolutional Neural Networks (CNNs)** have been demonstrated to be more accurate for image classification tasks than other feature-based automated methods. This is due to the fact that CNNs, with respect to Feed-Forward Neural Networks (FFNN), are able to keep the information of the correlations between adjacent pixels in the 2D space. It is demonstrated that reliable classifications can be obtained using CNNs even with small datasets, with an adapted training strategy (Huertas-Company and Lanusse 2023).

4. Results

Given the motivations above, we chose to create a CNN in order to classify the galaxies in our dataset. We followed a multi-dimensional regression approach in order to replicate the set of target labels. Hence, we used the Root Mean Square Error (RMSE) as loss function, following the guidelines of the Kaggle challenge.

We split the labeled images into a train, validation and test set. Then, we built our architectures using the Python libraries **PyTorch** and **TorchVision**. In particular we used the Pytorch class `torch.nn.Module`. In order to train it, each network required a `DataSet` and a `DataLoader` class, which loaded the images in greyscale with their labels, organized them in batches, converted each one into a PyTorch tensor and executed 3 transformations to it:

- Cropping the borders of the image, reducing its size to 324×324 pixels;
- Resizing it into 128×128 pixels image;
- Applying a random rotation.

All architectures employ batch normalization via PyTorch's `BatchNorm2d` class, following the approach introduced by Ioffe and Szegedy (2015). Batch normalization should enable the use of higher learning rates and reduce sensitivity to weight initialization. Additionally, it should obviate the need for bias parameters in convolutional layers and, in some cases, eliminate the need for dropout.

Lastly, we implemented a function whose task is to remap the original 37 labels into 17 independent classes following the Hubble–de Vaucouleurs morphological classification. Each of the resulting labels corresponds to the independent probability that the galaxy belongs to a specific morphological class. This set of labels is by construction always normalized to one. We reduced the Hubble-de Vaucouleur classification to a simplified version according to the complexity degree of the Galaxy Zoo decision tree. For example edge-on galaxies are managed by only three answers, and so we condensed similar morphologies into three corresponding main classes:

- lenticular/early-type spiral (*S0/Sa edge-on*);
- barred lenticular/early-type barred spiral (*S0B/SBa edge-on*);
- late-type spiral (*Sc/d edge-on*).

In a similar way, the classes regarding elliptical galaxies were reduced into three types, spacing from early-type to late-type with the symbolic labels *E0, E3* and *E6*. Finally, we did not take into account irregular classes and added the class 'Artifact' to deal with spurious images in the dataset.

The overall workflow consisted of designing and evaluating multiple CNN architectures. For each one, we performed a systematic hyperparameter optimization using the **Optuna** framework, exploring a defined hyperparameter space including:

- **Weight initialization**, selecting between He initialization [3] and PyTorch's default initialization;
- **Optimizer**, with candidates comprising RMSprop, stochastic gradient descent (SGD), and several variants of the Adam optimizer (e.g., AdamW, Nadam, Adagrad);
- **Momentum** between 0.3 and 0.9 with 0.1 steps for RMSprop and SGD.
- **Activation function**, comparing ReLU and Leaky ReLU;
- **Learning rate** as a continuous parameter in the range between 10^{-1} and 10^{-5}.

Each hyperparameter study was constrained to a maximum of 50 training epochs to balance computational cost with optimization depth. Following this phase, we selected the best-performing configuration–defined by validation performance–and resumed training of the corresponding model until overfitting.

We ran our models both on our personal laptops and on a dedicated Virtual Machine on Cloudveneto[4], an infrastructure designed by the University of Padova and the Italian National Institute of Nuclear Physics (INFN) for scientific research.

4.0.1. CNN configurations

We tuned the hyperparameters for 5 different structures of CNNs, followed by training for a sufficient number of epochs until the curve of the validation loss was flattened. In this section we summarize the main features of these architectures. The details of each network are shown in visual schemas in Appendix A. Every network is composed of a number of convolutional layers, followed by a block of fully connected layers.

The first network, called **JAGZoo**, is characterized by 6 blocks. Each of the first five comprehends a convolutional layer, an activation function, and then BatchNorm and Max Pooling layers. The sixth block is composed of 3 fully connected layers. We also implemented a version of this architecture that trains directly on the mapped features and inverts the activation and the BatchNorm layer for each block.

The second architecture, called **PADel**, is slightly more complex. It is composed of 7 convolutional blocks. They are similar to those contained in JAGZoo except for the MaxPooling layer

[3] He initialization (He et al. 2015) is specifically designed to maintain variance across layers with ReLU activation, and has been shown to outperform both Xavier initialization (Glorot and Bengio 2010) and the standard Gaussian initialization provided by PyTorch in terms of convergence speed and training stability.

[4] `https://cloudveneto.it/`

which is not applied in every block. The first four also apply 'same' padding: PyTorch computes a padding value to match the channels' output size to the input size, allowing deeper convolutional networks. This structure was also trained including the mapping function.

The third architecture, called **PC**, is the simplest model since it has only one convolutional block. The latter is structured in the same way as the blocks composing the JAGZoo network.

The fourth architecture is **SkyNet**. It consists of seven convolutional blocks, with batch normalization applied only to the first and the last three layers. Max pooling is applied exclusively to the fifth and the sixth layer. The blocks are followed by four fully connected layers with a large number of neurons. We set 0.5 dropout after the first and before the output layers.

Finally, we trained an architecture that recalls the reference model **VGG16** (Simonyan and Zisserman 2015). It differs in the implementation of BatchNorm after each activation layer, and a smaller fully connected block.

The parameters tuned by Optuna in the first 50 epochs were used for training of the remaining epochs and are shown for each architecture in Table 1.

Name	Weight init	Optimizer	Momentum	Activation	Learning rate
JAGZoo	False	NAdam	None	LeakyReLU	0.001
JAGZoo-m	False	Adam	None	ReLU	0.01
PADel	False	NAdam	None	LeakyReLU	0.0007
PADel-m	False	RMSProp	0.8	ReLU	9×10^{-5}
PC	False	SGD	0.6	ReLU	0.002
SkyNet	False	Adam	None	ReLU	0.0002
VGG	False	RMSProp	0.6	ReLU	3×10^{-5}

Table 1: Values of the parameters for each architecture found after optimization. The alternative networks which implemented mapping are referred to using the suffix '-m'.

4.0.2. Architecture results

We show in Figure 2 the performances of our models per epoch.

We saved the models' state at an epoch where the trend of the validation loss flattens and calculated the loss on the test set. We report in Table 2 the test losses. We calculated them for both the 37-dimensional labels (the original labels), and the 17-dimensional labels (the ones obtained after mapping the values). We chose PADel and VGG as the best performing models.

Model	Test Loss 37	Test Loss 17
JAGZoo	0.0933	0.0644
JAGZoo-m	/	0.0613
PADel	0.082	0.0604
PADel-m	/	0.0614
PC	0.108	0.0780
SkyNet	0.0860	0.0603
VGG	0.0850	0.0596

Table 2: Summary of the performances of the different architectures. The test losses are calculated at the epoch where the validation loss flattens. We show the losses for the 37 labels in the original format, and the 17 labels obtained after mapping them in the morphology classes.

Using the set of 17 mapped labels, we produced a *true label* and a *predicted label* for each image. They correspond to the morphology classes that had the maximum probability from both the test label set and the one computed by the CNN. Furthermore, we selected the images whose morphology class for the true label was 'clear' even to human classifiers, meaning that the distance between the highest and the second highest probability on the test labels was greater than 0.1. Having a direct comparison between the true label and the predicted label, we were able to produce the fraction of correctly classified galaxies for each morphology class. For each of these we also show the number of images present in the sample versus the number of images which are misclassified. The results obtained for PADel and VGG are shown in Table 3 (the other CNNs' scores can be consulted in Appendix B). In Figure 3 we show the confusion matrices for both CNNs.

True Label	Samples	PADel		VGG	
		Misclassified	Score	Misclassified	Score
E0	1064	48	0.95	43	0.96
E3	1182	138	0.88	130	0.89
E6	170	48	0.72	35	0.79
S0A	698	385	0.45	350	0.50
S0B	119	54	0.55	38	0.68
SAa	1	1	0.0	1	0.0
SAb	142	53	0.63	61	0.57
SAc	337	59	0.82	80	0.76
SAd	5	5	0.0	5	0.0
SBb	65	30	0.54	30	0.54
SBc	66	27	0.59	24	0.64
SBd	1	1	0.0	1	0.0
S0/a e/o	292	52	0.82	64	0.78
Sc/d e/o	111	32	0.71	33	0.70
SB0/a e/o	2	2	0.0	2	0.0
A	6	0	1.0	1	0.83

Table 3: Results for PADel and VGG. The morphology classes follow the Hubble-de Vaucouleurs schema. In the second column are shown the total number of samples for each category. In the 'Misclassified' columns are shown the number of images which were misclassified by the algorithm for each class. The 'Score' columns show the fraction of correctly classified galaxies for each morphology category.

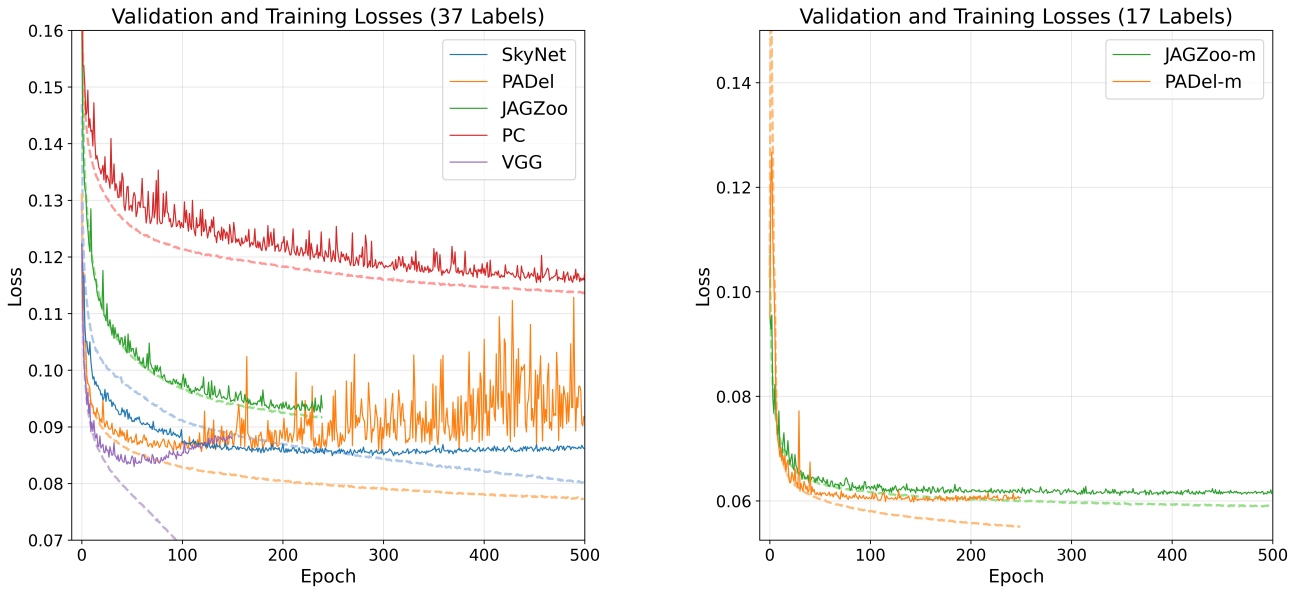
Eventually, we used the PADel network to classify a new set of unlabelled images included in the Galaxy Zoo dataset. We show in Figure 4 some examples of galaxies that the CNN classified as early-type ellipticals (*E0*) and late-type spirals (*SAc*).

5. Discussion

As can be seen in Figure 2, most of the architectures' training loss curves would dramatically decrease in steepness after $\sim$ 50 epochs leading to a flat validation curve. This behaviour becomes less and less pronounced the deeper the network is, hinting that the simpler ones stopped learning valuable features and focused on learning noise-level features.

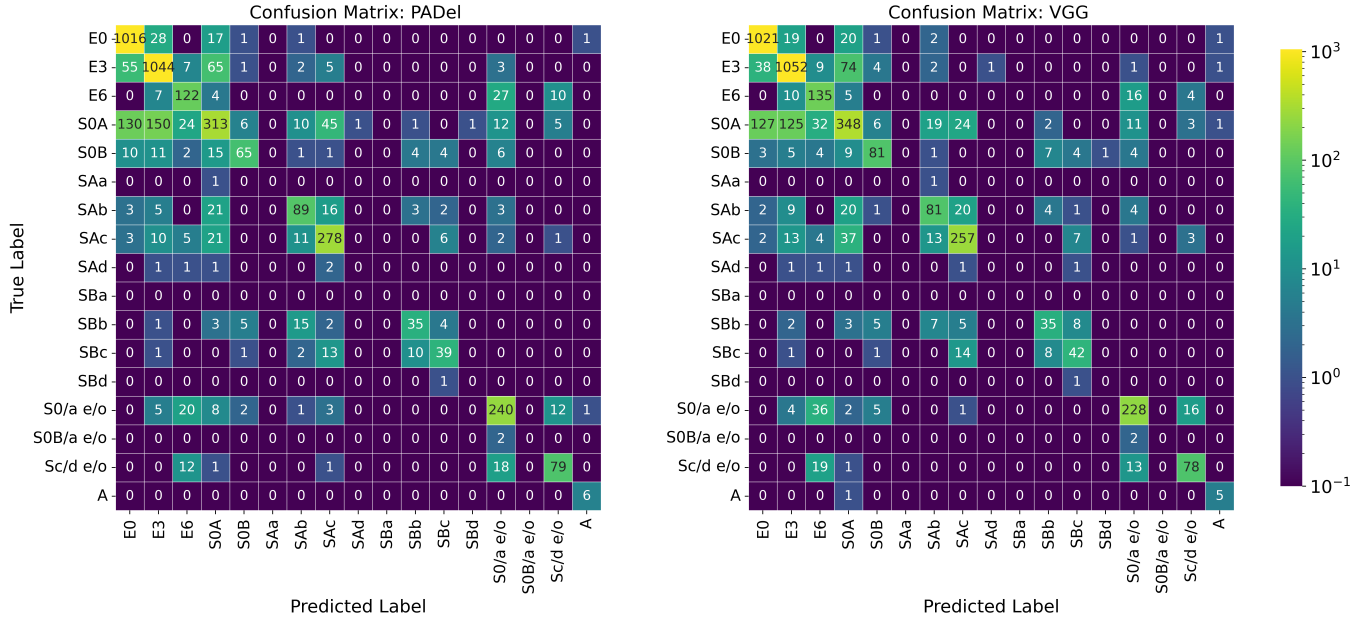
In Table 3 and Figure 3 we observe that the two best models perform similarly. The scores are affected by morphology classes themselves: elliptical galaxies are the easiest to classify, while the CNNs struggle to coherently recognise lenticular galaxies and fainter features such as the bar in spiral galaxies. Lenticular galaxies in particular are often classified by the model as elliptical, which is understandable given the similarities between the two.

The deepest models–namely Skynet, PADel, and VGG–are the best performing ones, yielding similar scores. PADel could be the best choice, since in terms of computational cost and time it is the least demanding of the three.



(a) (b)

Fig. 2: Training (dashed lines) and validation (solid lines) losses for the trained models.

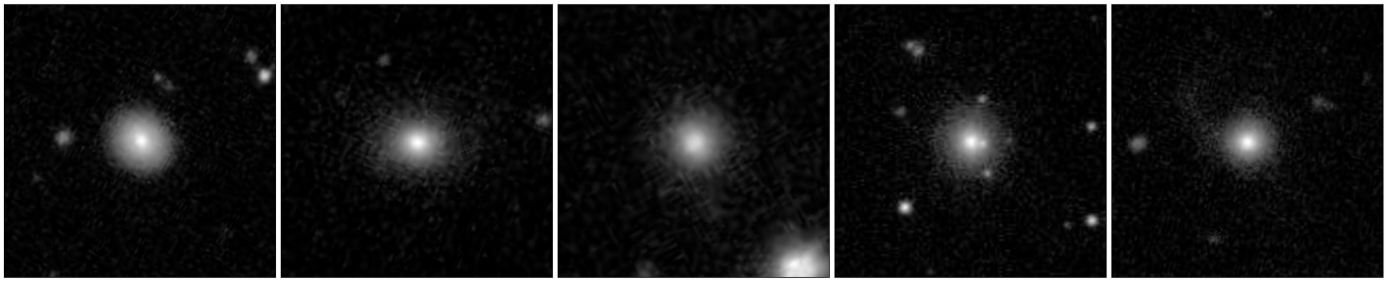


(a) (b)

Fig. 3: Confusion matrices for PADel and VGG. In each row is summarized how the images from each class are classified by the CNN. In particular, the diagonal contains the correctly classified ones.

6. Conclusions

In this work, we took into consideration the Galaxy Zoo Challenge Dataset, a collection of galaxy images provided with a set of labels that refer to a classification process applied to them using public answers on the Kaggle platform. We applied a Deep Learning approach, since it has become the state-of-the-art approach for the classification of large datasets for its performance, velocity and its ability to scale well with the size of the dataset. We built and trained a variety of Convolutional Neural Networks (CNNs) to perform a regression on the label set of the images. We then mapped these labels into a set of independent probabilities associated with a set of morphology classes inspired by the Hubble-De Vaucouleurs classification. We presented, for the two best models, the fraction of correctly classified galaxies for each morphology class, along with their confusion matrices. Our networks were able to distinguish well the elliptical and spiral galaxies and to divide them into early and late-type categories.



(a) Example of E0 galaxies found by our network.

(b) Example of SAc galaxies found by our network.

Fig. 4

It performed worse in recognising lenticular galaxies and faint features like bars.

There are many further possibilities that could be explored in continuity with this work. We could keep studying CNNs structures and training procedures by trying to vary parameters such as kernel size, stride, dilation and padding. We could test if there would be any change in inputting the images in RGB, as the colour of galaxies tends to relate with their morphology. Furthermore, we could try different map functions that include new morphological classes such as irregulars. Another possibility could be implementing the standardization of the dataset in order to improve the model performance by getting more equally weighted classes.

References

Buta, R. J. (2011). Galaxy morphology.

Dieleman, S., Willett, K. W., and Dambre, J. (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450(2):1441–1459.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.

Hubble, E. (1936). Realm of the nebulae. yale university press, new haven.

Huertas-Company, M. and Lanusse, F. (2023). The Dawes Review 10: The impact of deep learning for the analysis of galaxy surveys. *PASA*, 40:e001.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. 37:448–456.

Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., Murray, P., and Vandenberg, J. (2008). Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *MNRAS*, 389(3):1179–1189.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition.

Willett, K. W., Lintott, C. J., Bamford, S. P., Masters, K. L., Simmons, B. D., Casteels, K. R. V., Edmondson, E. M., Fortson, L. F., Kaviraj, S., Keel, W. C., Melvin, T., Nichol, R. C., Raddick, M. J., Schawinski, K., Simpson, R. J., Skibba, R. A., Smith, A. M., and Thomas, D. (2013). Galaxy Zoo 2: detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey. *MNRAS*, 435(4):2835–2860.

Appendix A: CNN architectures

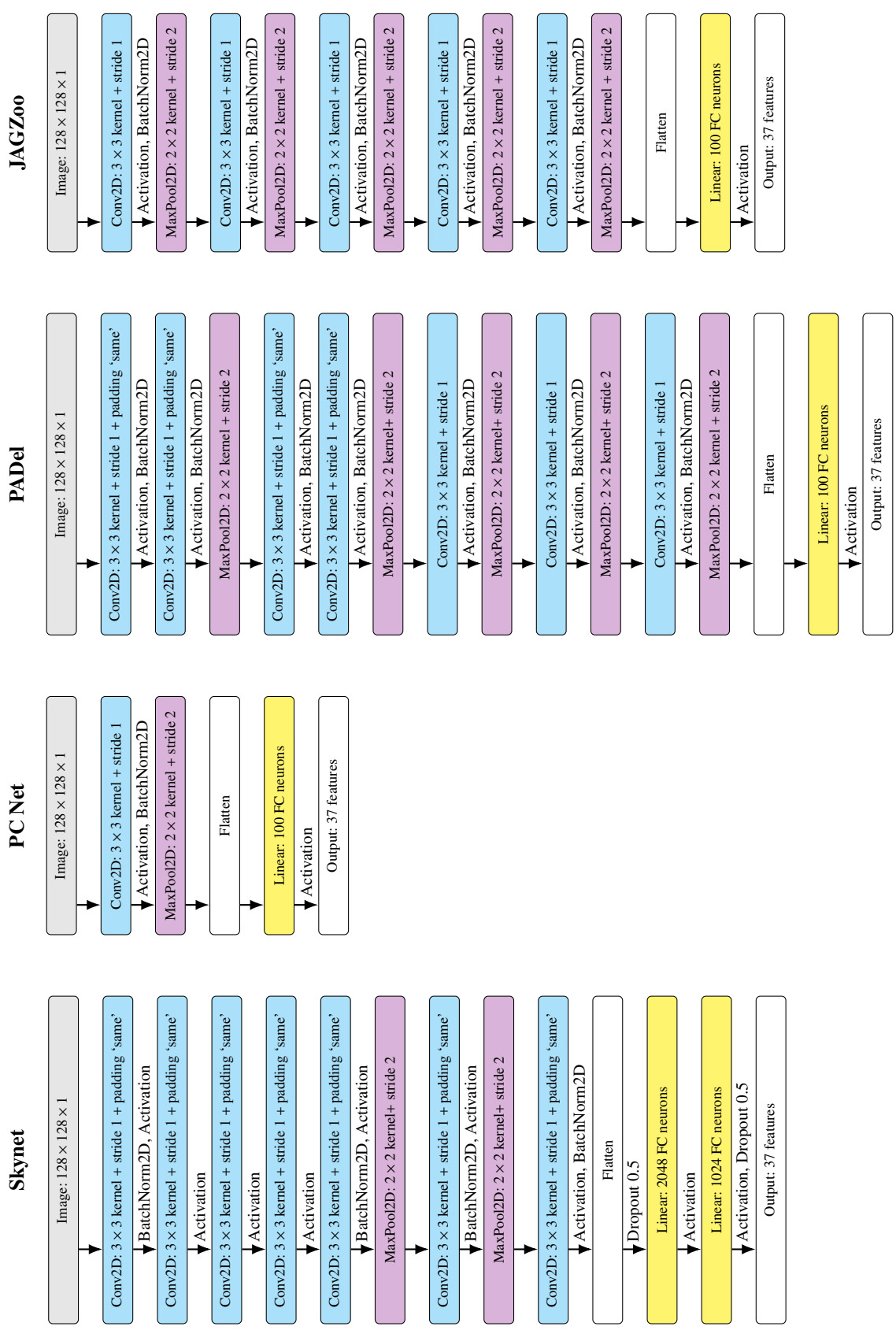


Fig. A.1: Diagrams of the architectures. '-m' versions differ in applying the map function on the output layer.

Appendix B: Results

True Label	Samples	JAGZoo		JAGZoo-m		PADel-m		SkyNet		PC	
		Misclassified	Score	Misclassified	Score	Misclassified	Score	Misclassified	Score	Misclassified	Score
E0	1064	36	0.97	52	0.95	59	0.94	55	0.95	40	0.96
E3	1182	151	0.87	118	0.90	138	0.88	142	0.88	474	0.6
E6	170	48	0.72	51	0.70	44	0.74	49	0.71	155	0.09
S0A	698	410	0.41	390	0.44	353	0.49	366	0.48	491	0.3
S0B	119	60	0.50	48	0.60	42	0.65	45	0.62	94	0.21
SAa	1	1	0.0	1	0.0	1	0.0	1	0.0	1	0.0
SAb	142	86	0.39	73	0.49	71	0.5	63	0.56	131	0.08
SAc	337	109	0.68	80	0.76	76	0.77	67	0.80	166	0.51
SAd	5	5	0.0	4	0.20	5	0.0	4	0.20	5	0.0
SBa	0	-	-	-	-	-	-	-	-	-	-
SBb	65	30	0.54	30	0.54	30	0.54	27	0.58	46	0.29
SBc	66	38	0.42	24	0.64	25	0.62	19	0.71	58	0.12
SBd	1	1	0.0	1	0.0	1	0.0	1	0.0	1	0.0
S0/a e/o	292	78	0.73	49	0.83	47	0.84	50	0.83	69	0.76
Sc/d e/o	111	27	0.76	29	0.74	28	0.75	27	0.76	33	0.7
SB0/a e/o	2	2	0.0	2	0.0	2	0.0	2	0.0	2	0.0
A	6	5	0.17	0	1.0	1	0.83	1	0.83	4	0.33

Table B.1: Results for the remaining five architectures. The morphology classes follow the Hubble-de Vaucouleurs schema. The 'Samples' column shows the number of samples for each type of galaxy. For each architecture, the first column shows the number of misclassified galaxies, while the second column shows the score of the architecture in predicting that specific type of galaxy.