Nonlinear Analysis: Modelling and Control, Vol. 31, No. 3, 544–561
https://doi.org/10.15388/namc.2026.31.45923

Vilnius University Press

Evaluation of domain-specific convolutional neural network layers for galaxy classification tasks

Tomas Mūžas, Andrius Vytautas Misiukas Misiūnas, Tadas Meškauskas

Vilnius University, Faculty of Mathematics and Informatics,
Institute of Computer Science
Didlaukio 47, Vilnius LT-08303, Lithuania
tomas.muzas@mif.vu.lt

Received: May 29, 2025 / **Revised:** December 23, 2025 / **Published online:** March 12, 2026

Abstract. Due to increasing amount of data, galaxy image classification is being automated using machine learning, with the most common approach being convolutional neural networks (CNNs). Using domain-specific observations about galaxy morphology, we propose two CNN layers – Focal and Wave. The Focal layer highlights information in the center, while the Wave layer is designed to highlight spiral arms. A comprehensive evaluation of layers is performed using datasets of 4 and 11 (not yet deeply studied) galaxy types, with and without augmentations. With no augmentations, our models outperform best results of models used in other studies with Wilcoxon signed-rank test's p-value of $p_{\mathrm{F1}} = 0.002$ for both 4 and 11 classes in terms of F1 score (used to better evaluate performance due to inherent class imbalance). Moreover, our layers help reduce overfitting and reliance on augmentation combinations, as our models, despite benefiting less from augmentations, still achieve results comparable to other augmented models. Additionally, our models tuned for 4-class data perform equally well on 11-class data, and vice versa. Finally, an ensemble of our models trained without augmentations achieves comparable results to our augmented models.

Keywords: domain-specific layers, galaxy morphology classification, Galaxy Zoo, model tuning, convolutional neural networks.

1 Introduction

Based on the visual appearance, or morphology, galaxies are grouped into two major categories – spiral and elliptical. Spiral galaxies have spiral arms that extend from the center of the galaxy. They contain active star-forming regions and host younger, bluer stars. Elliptical galaxies are round and smooth, with less star-forming activity, containing older, redder stars.

Those types of galaxies can be further subdivided. One such classification scheme is the Hubble Tuning Fork proposed by Hubble in 1926 [7]. In this scheme, spiral galaxies

are split on the presence of bars – barred (SB) and unbarred (S), which are then further classified by the tightness of their arms, assigning letters a–d. For example, SBa is a barred spiral galaxy that has tight arms. Elliptical (E) galaxies are classified by their elongation, assigning a number $0-7$. A fully round elliptical galaxy would be classified as E0.

Initially, galaxies were classified visually inspecting telescopic images. Some larger datasets were produced this way, such as Nair and Abraham [15], containing approximately 14,000 galaxies. As the amount of data grew, manual approach was superseded by digital volunteer classification project Galaxy Zoo (GZ) [12], in which volunteers were presented with an image of a galaxy and had to answer visual questionnaires, resulting in approximately 900,000 classifications. Following the success of this project, a multitude of new projects was developed – GZ 2 [26], GZ CANDELS [18], GZ DECaLS [21], GZ Hubble [25], GZ DESI [20]. However, as more surveys are launched, containing hundreds of millions of images, such as DES [1], it becomes difficult to collect enough volunteer votes to keep up with the upcoming amount of data.

Therefore, the latest approach is to automate galaxy classification using machine learning. While multiple machine learning methods have been studied, convolutional neural networks (hereinafter CNN) are studied the most, achieving some great results for spiral vs. elliptical classification – up to 99.5% [2]. However, due to the extreme distance and inherent class imbalance in nature (see Figs. 1 and 2), discerning different features becomes more difficult, especially when more nuanced classes are used – for 11 classes, the accuracy of current models drops to 65.2% [2] (see Section 2 for details).

To help models better distinguish galaxy features, we propose two novel layers for CNNs – Focal and Wave – that are based on galaxy domain-specific observations. The Focal layer is based on the fact that galaxies are in the center of the image, and background does not carry much useful information, hence it tries to minimize its weight. The Wave layer consists of multiple waves, going from the center of the image, which are intended to highlight any spiral arms or evaluate their tightness. Both layers are designed as masks that follow the regular convolution operation. For more information on layer implementation, see Section 3.2.

After conducting multiple tests with both 4- and 11-class datasets (see Section 3.1), both with and without augmentations (described in Section 4.2.2), the proposed Focal layer has been proven to be a useful improvement for CNNs for galaxy classification, as in the majority of cases, it is preferred during the model tuning process (see Section 4.1) over a regular convolution layer. Moreover, it is shown that this layer helps with regularization, as models containing this layer, trained without augmentations achieve better results than other augmented models. Finally, an ensemble of 4-class models trained with augmentations outperforms the overall best results achieved by an augmented Cavanagh et al. model.

The paper is structured as follows: in Section 2, related work is discussed. Section 3 describes data gathering and filtering process, as well as the implementation details of the proposed layers. Model tuning and training details are discussed in Section 4. The in-depth analysis and discussion of the results are in Section 5. Finally, conclusions and future work ideas are outlined in Section 6.

2 Related work

CNNs have been used in various galaxy classification tasks. However, before discussing the results, it is crucial to highlight that it is difficult to directly compare the results between authors, as there are currently no benchmark datasets or agreed classes, and each author chooses different datasets or subsets of GZ data.

One of the most studied problems in terms of galaxy classification is binary classification into spiral and elliptical. Near-perfect results have already been achieved in this field: on the brightest galaxies, Barchi et al. [2] have achieved 99.5% accuracy using GoogleNet Inception [19]. Cheng et al. used their own CNN architecture to achieve impressive results as well – 99.4% accuracy [5].

Classification into more classes has also been studied extensively. In terms of 3 classes, classification results are still close to perfect, such as 98% achieved by Mittal et al. using a small GZ1 data subset [13] or 97.7% achieved by Khalifa et al. [9]. As for 5 classes, Zhu et al. [27] have achieved up to 95.2% accuracy.

Though the aforementioned study by Barchi et al. [2] highlights the overall trend when comparing the models trained using the same conditions and data, but using different classification schemes. In their study, results for 3 (up to 82.7% accuracy), 7 (up to 77.6% accuracy), 9 (up to 75.7% accuracy) and 11 classes (up to 65.2% accuracy) are presented, obtained using CNNs and other traditional machine learning methods. Those results show that classification into large amount of classes ($7-11$) is still difficult.

Another relevant aspect to this study is the usage of domain-specific knowledge when constructing CNNs. To the best of our knowledge, the layers which are specifically designed to reflect galaxy morphology and image placement (see Section 3.2) were not yet explored. However, there were several notable examples of using more generic, yet still morphology-aware layers. The first one being Wei et al. [22], who used Deformable Convolution Networks to help capture galaxy internal structure, achieving close to 95% F1 and accuracy scores for 7-class problem. Another prominent example is Capsule Networks to obtain rotation-invariant, spatially-aware features, used by Li et al. [11] and Katebi et al. [8].

3 Methodology

This section describes data gathering and processing for 4- and 11-class datasets (Section 3.1) and the implementation details of the proposed layers (Section 3.2).

3.1 Data

This section describes data collection, filtering, and processing steps. Two different processes were performed for 4- and 11-class classification.

Both datasets serve unique purposes in this study. As there is enough data, 4-class dataset is intended to test model performance in idealized conditions – all unreliably classified galaxies are removed, leaving only galaxies with the most certain and consistent classifications, which also implies better image quality. For details, see Section 3.1.1.

On the contrary, 11-class dataset is intended to test model performance on a difficult classification task, maintaining as large image variety as possible. The selection process is outlined in Section 3.1.2.

3.1.1 4 classes

For 4-class dataset, volunteer classification data from Galaxy Zoo (GZ) 2 [26], GZ CANDELS [18], GZ DECaLS [21], and GZ Hubble [25] was combined into a single dataset. GZ 1 dataset [12] was not included, as it only contained classifications into spiral and elliptical.

Since some of the galaxies in the aforementioned datasets overlap, it was first important to ensure that the galaxy class is consistent across different datasets. As galaxies have different IDs in different datasets, they were first cross-matched by their center coordinates. Two entries were considered the same galaxy if their centers were within 0.5 arcsec.

Once the galaxies were cross-matched, it was crucial to ensure that the galaxy is classified reliably. To evaluate that, first the classes were assigned using the following strict rules for galaxy flags:

For GZ2 and GZ DECALS data:

- *Elliptical* if `Smooth` $\geqslant 0.8$ and `Features` < 0.8.
- *Edge-on* if `Features` $\geqslant 0.8$, `Smooth` < 0.8, and `Features_EdgeOn_Yes` $\geqslant 0.8$.
- *Spiral Bar* if `Features` $\geqslant 0.8$, `Smooth` < 0.8, `Features_EdgeOn_No` $\geqslant 0.8$, and `Features_Flat_Bar_Yes` $\geqslant 0.8$.
- *Spiral No Bar* if `Features` $\geqslant 0.8$, `Smooth` < 0.8, `Features_EdgeOn_No` $\geqslant 0.8$, and `Features_Flat_Bar_No` $\geqslant 0.8$.

The flags `Smooth` and `Features` correspond to the first question in GZ questionnaire – `Smooth` means that the galaxy is round, smooth and has no notable features, thus being elliptical, while `Features` indicates that the galaxy has features, such as arms, making it spiral. After that, galaxies were further divided into edge-on (using `Features_EdgeOn_Yes/No` flag), barred (`Features_Flat_Bar_Yes`) and unbarred (`Features_Flat_Bar_No`). All galaxies that did not meet any of the aforementioned criteria, were assigned *Uncertain* class.

For GZ Hubble and GZ CANDELS data, the same rules as for GZ2 and GZ DECALS data were applied, though additionally applying rule `Features_Clumpy_No` $\geqslant 0.8$ for *Edge-On*, *Spiral Bar*, and *Spiral No Bar* classes to eliminate clumpy galaxies.

Once classes for each dataset were assigned, galaxies were required to have the same class in all datasets they are present. All occurrences of a galaxy that had at least one mismatching class were removed entirely. Additionally, all remaining galaxies with *Uncertain* class were removed as well.

Such a strict filtering process meant that out of almost 750,000 galaxies, only 99,334 (approximately 13%) unique galaxies were kept that met all the criteria. However, this process ensured that the galaxy is reliably classified. Nonetheless, it is important to

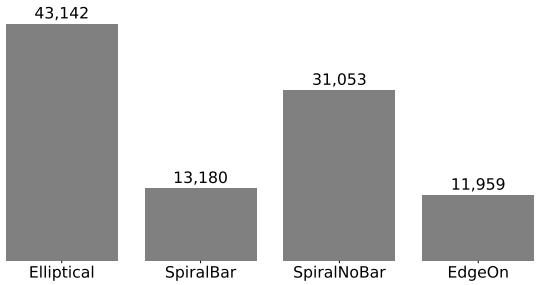

Figure 1. Class distribution of 4-class dataset.

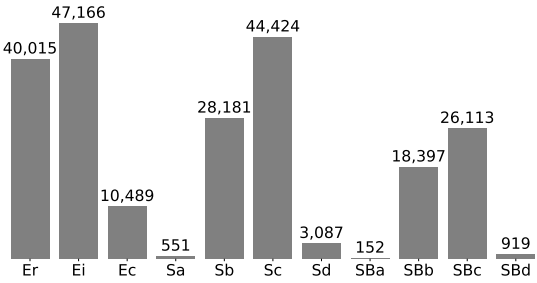

Figure 2. Class distribution of 11-class dataset.

mention that such dataset represents idealized data and it would not be a good approximation of real, diverse data. The data distribution is visualized in Fig. 1.

3.1.2 11 Classes

To produce the 11-class dataset, GZ2 dataset was used, along with the filtering process described by Barchi et al. [2], except for omitting the filtering by K-parameter to preserve as many galaxies as possible.

This meant that galaxies with the prefixes Er, Ei, Ec, Sa, Sb, Sc, Sd, SBa, SBb, SBc, and SBd were taken. No additional data processing was performed. This resulted in a dataset of 219,494 galaxies. However, the dataset is heavily imbalanced, which is visualized in Fig. 2. Although, it is important to mention that such data imbalance occurs naturally in the nature.

3.1.3 Images

The images for the GZ2 dataset were collected using SkyServer's ImgCutout[1] tool. For the remaining datasets, images available in Galaxy Datasets[2] were used. Images were originally obtained in size 424×424, cropped to 224×224 from the center, and then resized to a final size of 128×128.

[1] `http://skyservice.pha.jhu.edu/DR7/ImgCutout/getjpeg.aspx`
[2] `https://pypi.org/project/galaxy-datasets/`

3.2 Domain-specific layers

The motivation behind the domain-specific layers comes from the idea that when CNN models are intended to be used for galaxy classification only, it might be beneficial to utilize the specifics of the galaxy images to improve performance.

Having inspected galaxy images that were shown to the users in Galaxy Zoo projects, it is evident that the galaxy is in the middle of the image. This means that the center carries the most information, and the background is mostly black, or contains other stars or galaxies, which does not benefit classification. This particular observation inspired the Focal layer (Section 3.2.1), which aims to reduce the weight of the input further from the center.

Another layer – Wave – is based on the fact that galaxy arms extend from its center (in this case – from the image center as well). Additionally, the arms follow circular pattern. Therefore, the idea of the Wave layer is to project multiple spirals from the center to help facilitate detection of the spiral arm edge. The layer details are outlined in Section 3.2.2.

Both layers are designed to act as a mask for the output of convolution layers. It is important to note that the convolution operation is performed in sequence, and aggregates patches of images, thus producing a reduced version of the original input, though preserving the original order. Our masks are supposed to put additional emphasis on certain patches of the convolution outputs, which should be responsible for galaxy features. Finally, both layers are designed as 2D masks, thus they are expanded to the batch and depth dimensions of the output.

3.2.1 Focal layer

The Focal layer assigns the most weight to the center and gradually reduces the weights towards the edges of an input. The layer is produced using Eq. 1.

$$M = \exp\left[-factor \cdot \log 2 \cdot \frac{(x - x_0)^2 + (y - y_0)^2}{r^2}\right]. \tag{1}$$

Here $x_0 = y_0 = r = N/2$, where N is the size of an input. There is an additional tunable parameter $factor$, which controls the weight decay rate – the bigger the value, the faster the decay. The resulting masks with different $factor$ values are depicted in Fig. 3.

3.2.2 Wave layer

The second domain-specific layer is the Wave layer. This layer is designed as repeating waves, which should help better detect spiral arms of galaxies. Additionally, for the 11-class dataset, this layer might help estimate arm tightness (a–d in class names), thus potentially improving classification as well. The layer is produced using Eq. 2.

$$M = 1 - \cos\left(\frac{\pi}{k\_factor} \cdot \sqrt{(x - x_0)^2 + (y - y_0)^2}\right) \tag{2}$$

In Eq. 2, $x_0 = y_0 = N/2$, which represents the center of the input of size N. There is a tunable parameter k_factor which represents the frequency of the waves, the bigger the

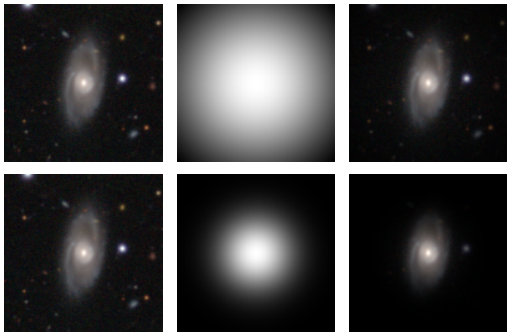

Figure 3. Illustration of the focal layer effect. The top row represents a Focal layer with $factor = 1$, bottom – $factor = 7$.

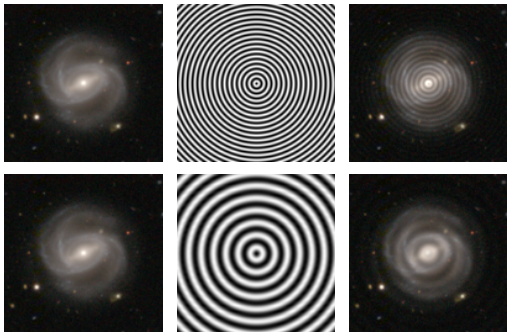

Figure 4. Illustration of the Wave layer effect. The top row represents wave layer with $k\_factor = 2$, bottom – $k\_factor = 5$.

value, the less waves are produced. The example of masks with different k_factor values are shown in Fig. 4.

4 Experiments

This section describes model tuning (Section 4.1) and training (Section 4.2) experiments and their setup in detail.

4.1 Model tuning

To evaluate the applicability of domain-specific layers proposed in Section 3.2, several model tuning iterations were carried out, for both 4- and 11-class datasets separately.

4.1.1 Tuning datasets

For both 4- and 11-class datasets, 10% of the initial dataset was reserved exclusively for tuning, ensuring that data distribution remains proportionally the same. The tuning

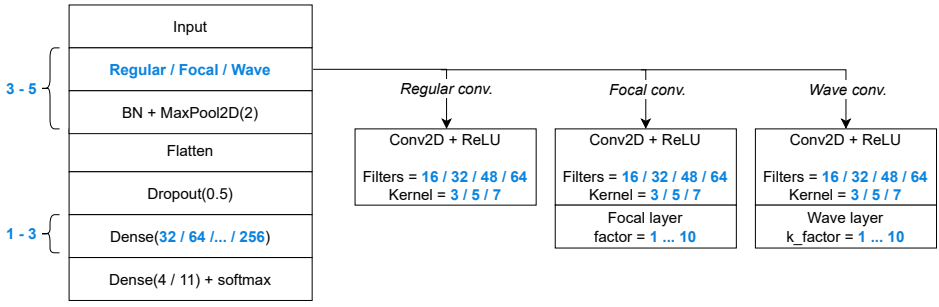

Figure 5. Model template for tuning. Text in blue denotes tunable hyperparameters and their possible values.

dataset was then further split into $90/10\%$ for training and test accordingly, using 10-fold stratified cross-validation method.

However, due to the different nature of the datasets, there are slight differences in galaxy selection. Since the 4-class dataset could have multiple images for the same galaxy from different GZ datasets, it was ensured that all images of one galaxy could belong only to the tuning split, or to the training split. The same was ensured for further splits into train and test subsets. For 11 classes, as there was only a single image per galaxy, such criteria was not required.

4.1.2 Base architecture

The architecture by Cavanagh et al. [4] is taken as a base, as it is lightweight and modular. The model template and all possible hyperparameter values are depicted in Fig. 5. In case of repeating convolution and dense layers, hyperparameter values were chosen independently for each layer.

4.1.3 Setup

The tuning setup for each of the datasets was as follows:

1. Tune models without augmentations for the best loss. Produced models:
   (a) Overall best model that achieved the best loss (hereinafter *Best Loss*)
   (b) Combination of top 10 best models, by taking median value for each of the hyperparameters (hereinafter *Top 10 Loss*).
2. Tune models without augmentations for the best F1 value, which better reflects model performance on imbalanced datasets (see Section 3.1). Produced models:
   (a) Overall best model that achieved the F1 value (hereinafter *Best F1*)
   (b) Combination of top 10 best models, by taking median value for each of the hyperparameters (hereinafter *Top 10 F1*).

The models were tuned using Keras Hyperband [16] tuner. The Hyperband bracket was 100 epochs, with a default factor of 3. The training for each hyperparameter combination was performed with Adam [10] optimizer, using default parameters, suggested by

the authors: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \cdot 10^{-8}$. The loss to optimize was categorical cross-entropy. The training process lasted at most for 100 epochs, but was stopped early if the loss value has not improved for 10 consecutive epochs.

4.1.4 Resulting models

In total, 8 models were produced – *Best Loss*, *Top 10 Loss*, *Best F1*, and *Top 10 F1* for both 4 and 11 classes separately. The simplified hyperparameter choices for these models are presented in Table 1, where parts (a) and (b) correspond to 4 and 11 class, respectively.

Even though the resulting models are slightly different, some common observations can be made from Table 1. First of all, for both datasets, better loss values are achieved by bigger models – both for *Best Loss* and *Top 10 Loss*, 4 convolution layers are chosen. As for the best F1 value however, the best results are achieved with smaller models, consisting of 3 convolution layers, which is also consistent for both datasets.

Another important observation is that for both datasets, the new Focal convolution layers are preferred over regular and Wave convolutions, as indicated by the architectures of *Top 10 Loss* and *Top 10 F1* models of both datasets. Higher $factor$ values, ranging from 5 to 10, are preferred. However, the Wave convolution was occasionally chosen as well, although only once, for both *Best Loss* and *Top 10 Loss* models of 4 classes, and *Best F1* model of 11 classes.

Finally, even though there was an opportunity to not apply any of our layers, inspecting the top 10 most popular choices for both datasets and both metrics, regular

Table 1. Simplified hyperparameter choices of the best models. For convolutions, first number is number of channels, second is kernel size, and for Focal convolution, third number is $factor$ or k_factor for Wave convolution.

Model	Convolutions	Dense
	(a) 4 classes	
Best Loss	Focal (64/7/8), Regular (64/7) Regular (48/3), Wave (64/5/5)	224/224/160
Top 10 Loss	Focal (64/7/8), Focal (64/5/5) Regular (32/7), Wave (3/5/3)	32/160/160
Best F1	Focal (16/7/7), Focal (48/7) Focal (32/7/6)	224/128/64
Top 10 F1	Focal (16/7/5), Focal (48/7/3) Focal (32/7/6)	192
	(b) 11 classes	
Best Loss	Focal (48/3/10), Regular (32/7) Regular (48/7), Regular (32/3)	224/192
Top 10 Loss	Focal (48/7/10), Focal (32/7/7) Focal (32/5/8), Focal (32/5/6)	128/96/64
Best F1	Focal (16/5/6), Focal (64/3/9) Wave (48/7/2)	192/160
Top 10 F1	Focal (16/7/6), Focal (32/5/6) Focal (48/7/5)	128/160

convolution was chosen only for one layer, in 4-class *Top 10 Loss* model. Therefore, it can be concluded that our proposed improvements to convolution layers are chosen over the regular convolution, with Focal convolution being the most popular choice.

4.2 Model training

4.2.1 Setup

After tuning models for the best metrics as described in Section 4.1, the resulting models were trained on the training splits of 4- and 11-class datasets. The models were also compared against CNN models from other studies that achieved great results. The first model was from Cavanagh et al. [4], whose architecture was used as a base for tuning. Their architecture is an improved version of the Dieleman et al. [6] architecture. The second architecture was a similar and slightly simpler one used by Cheng et al. [5]. The training process was the same as the tuning process described in Section 4.1.3, except that the models were allowed to train for up to 500 epochs instead of 100.

4.2.2 Augmentations

In order to test model performance in as many different and realistic scenarios as possible, it is important to evaluate the impact of augmentations, as they help to reduce overfitting, especially given the highly imbalanced data used in this paper (see Section 3.1). Several augmentation techniques were used to compare the results of models trained with and without augmentations. The techniques were as follows:

1. Rotating image by 90, 180, or 270 degrees, also used by multiple authors [4–6, 9, 13, 27].
2. Flipping image horizontally and vertically, also used by [4, 6, 9, 13, 27].
3. Zooming into image by cropping 10 pixels from each side of an image and rescaling it back to the size of 128×128 pixels. This technique or its variations were used by [6, 9, 27].
4. Random noise, which was generated by sampling values from -0.2 to 0.2 from the uniform distribution. The value of each pixel was later clipped to be between 0 and 1. This technique was used by [3, 5].

It is worth noting that in order to fully test the impact of specific augmentations to the proposed layers, all methods separately and multiple combinations of them should be tested, as some may be more impactful than others, or even have a negative effect. However, due to the limited computational resources, the aforementioned augmentation techniques were used in a single combination, as they provided the most stable results in our previous study [14].

As the datasets are quite large (see Sections 3.1.1 and 3.1.2), the augmented images were not precomputed, but rather applied on the fly, with an independent probability of 0.5 for each method. Hence, several, or in rare cases even all or none, methods could have been applied on a single image. Augmentations were not applied during the test phase.

5 Results and discussion

In this section, the results of model training for 4- and 11-class data are discussed.

5.1 Model comparison with class imbalance

It is important to note that since both datasets were imbalanced (see Section 3.1), the most important metric highlighted throughout the section is F1. Accuracy score is not trustworthy in such case, especially for 11-class dataset, where classes such as Sa or SBa have very few samples. However, accuracy, along with precision and recall, were included since it is frequently reported by other authors. Additionally, accuracy score allows to compare results with Barchi et al. [2], as it was the only reported metric. F1, precision and recall metrics all use macro averaging throughout the section.

To properly compare model performance, the p-value of the Wilcoxon signed-rank test [24] was used. If $p < 0.05$, null-hypothesis (median of the differences is close to 0) can be rejected, thus proving that there is a statistically significant difference between the metrics. As highlighted above, only F1 metric will be used for comparison, as it better reflects performance with imbalanced datasets. P-value calculated for F1 metrics will be denoted p_{F1} throughout the paper.

It is important to note that other tests could have been used. For example, Welch's t-test [23], a generalization of the Student's t-test that does not assume variances to be equal or paired t-test, as the metric values across folds are related. However, both of those tests assume that the populations are normally distributed. While the populations of F1 metrics did pass the Shapiro–Wilk [17] test with values over 0.75, the power of the metric for a population of 10 samples (folds) is low. Therefore, it was decided to use Wilcoxon's signed-rank test [24] as it does not assume a normal distribution. However, the results were compared using the other two tests as well, without any differences in outcome. The p-values of all three tests, along with the 95% confidence interval of F1 metric (denoted CI_{F1}), will be provided for context for the most important claims, while only Wilcoxon's p-value will be used elsewhere.

5.2 4-class results

The results of 4-class models trained without augmentations are shown in Table 2(a), while results with augmentations are shown in Table 2(b).

According to Table 2(a), it can be observed that when training without augmentations, all our models significantly outperform Cheng et al. [5] and Cavanagh et al. [4] models. The overall best results without augmentations were achieved by *Top 10 Loss* model, as it has the best overall F1 and accuracy scores.

Comparing F1 scores between our *Top 10 Loss* model and the best model by other authors, Cavanagh et al. [4], Wilcoxon test's p-value $p_{\mathrm{F1}} = 0.002$ is significantly less than 0.05, meaning that there is a statistically significant difference, or improvement between metrics. For comparison, paired t-test's $p_{\mathrm{F1}} = 1.7 \cdot 10^{-5}$, Welch's $p_{\mathrm{F1}} = 1.2 \cdot 10^{-5}$, and our 95% confidence interval $CI_{\mathrm{F1}} = (87.09, 87.84)$ not overlapping with their $CI_{\mathrm{F1}} = (85.38, 86.34)$, all strongly agree with statistically significant improvement. As for Cheng

Table 2. Results for 4 class dataset. Overall best results are highlighted in bold.

Model	F1 (σ)	Accuracy (σ)	Precision (σ)	Recall (σ)
	(a) Results without augmentations			
Cheng	82.89 (0.44)	83.85 (0.27)	83.71 (0.66)	82.37 (0.62)
Cavanagh	85.86 (0.63)	86.09 (0.55)	86.98 (1.14)	85.00 (0.71)
Best Loss	87.36 (0.57)	87.37 (0.52)	87.88 (1.03)	**86.99 (0.51)**
Top 10 Loss	**87.46 (0.50)**	**87.49 (0.42)**	88.08 (0.74)	86.99 (0.57)
Best F1	87.09 (0.47)	87.19 (0.43)	88.10 (0.58)	86.27 (0.69)
Top 10 F1	**87.46 (0.50)**	87.43 (0.53)	**88.12 (0.78)**	86.92 (0.46)
	(b) Results using augmentations			
Cheng	88.16 (0.40)	88.20 (0.38)	88.93 (0.32)	87.54 (0.53)
Cavanagh	**89.25 (0.20)**	**89.02 (0.21)**	**89.92 (0.35)**	**88.69 (0.31)**
Best Loss	89.04 (0.48)	88.82 (0.46)	89.66 (0.50)	88.52 (0.47)
Top 10 Loss	88.55 (0.35)	88.42 (0.34)	89.23 (0.59)	88.01 (0.37)
Best F1	88.92 (0.41)	88.74 (0.35)	89.63 (0.34)	88.33 (0.51)
Top 10 F1	88.92 (0.50)	88.74 (0.50)	89.64 (0.57)	88.32 (0.54)

et al. [5] model, the performance improvement is even more statistically significant: Wilcoxon $p_{\mathrm{F1}} = 0.002$, paired t-test $p_{\mathrm{F1}} = 9.9 \cdot 10^{-9}$, Welch $p_{\mathrm{F1}} = 5.9 \cdot 10^{-14}$, and $CI_{\mathrm{F1}} = (82.56, 83.22)$.

It is also important to highlight that our models outperform models of Cavanagh et al. [4] architecture that was used as the tuning template, showing that there is a tangible difference when adding our proposed layers.

Nonetheless, based on the Table 2(b), one important observation can be made. Cheng et al. [5] model has benefited the most from the application of augmentations, gaining approximately 5% increase for all metrics, while our models only improved by approximately $1{-}1.5\%$. However, even though our models benefited less from augmentations, they still achieved similar results to the best Cavanagh et al. [4] model ($p_{\mathrm{F1}} = 0.13$ comparing with our *Best Loss* augmented model), thus it can be implied that the proposed layers already act as a form of regularization and help reduce overfitting, even without applying augmentations.

Applying our layers could also help simplify training process, as our models already achieve higher performance of augmentations, without the need to select an appropriate augmentation techniques (Section 4.2.2) or their combinations.

5.3 11-class results

The results for 11-class training are shown in Table 3(a) (no augmentations) and (b) (with augmentations).

Based on Table 3, the observations are similar to 4-class results (see Section 5.2). When looking at the 11-class results without augmentations, our models again outperform others by a statistically significant amount. Comparing our *Best Loss* model with Cavanagh et al. [4], Wilcoxon's $p_{\mathrm{F1}} = 0.002$, paired t-test's $p_{\mathrm{F1}} = 8.0 \cdot 10^{-5}$, Welch's $p_{\mathrm{F1}} = 7.4 \cdot 10^{-6}$, and our 95% confidence interval $CI_{\mathrm{F1}} = (44.73, 46.32)$ not overlapping with their $CI_{\mathrm{F1}} = (42.01, 43.34)$. As for Cheng et al. [5] model, the same Wilcoxon's

Table 3. Results for 4-class dataset. Overall best results are highlighted in bold.

Model	F1 (σ)	Accuracy (σ)	Precision (σ)	Recall (σ)
	(a) Results without augmentations			
Cheng	40.81 (1.05)	60.71 (0.50)	45.74 (3.17)	40.02 (1.14)
Cavanagh	42.68 (0.88)	61.80 (0.42)	46.59 (2.31)	42.09 (1.08)
Best Loss	**45.52 (1.06)**	**63.79 (0.78)**	48.65 (3.46)	**44.76 (1.25)**
Top 10 Loss	44.63 (0.80)	63.43 (0.40)	47.88 (1.45)	43.63 (0.96)
Best F1	45.08 (0.80)	63.48 (0.30)	49.07 (1.17)	43.91 (1.06)
Top 10 F1	44.85 (0.84)	63.35 (0.33)	**49.70 (1.30)**	43.71 (0.90)
	(b) Results using augmentations			
Cheng	46.06 (1.05)	64.78 (0.18)	50.40 (1.56)	44.71 (1.11)
Cavanagh	**46.96 (0.67)**	**65.29 (0.33)**	**51.48 (2.07)**	**45.81 (0.86)**
Best Loss	46.04 (1.02)	64.65 (0.64)	49.21 (1.09)	44.83 (1.13)
Top 10 Loss	44.70 (1.28)	63.99 (0.77)	49.93 (2.61)	43.30 (1.23)
Best F1	46.04 (0.32)	64.83 (0.23)	51.20 (2.54)	44.65 (0.35)
Top 10 F1	44.88 (1.18)	64.25 (0.55)	50.21 (1.91)	43.59 (1.11)

$p_{\mathrm{F1}} = 0.002$ is achieved, though paired t-test's $p_{\mathrm{F1}} = 2.6{\cdot}10^{-7}$, Welch's $p_{\mathrm{F1}} = 2.0{\cdot}10^{-8}$, and $CI_{\mathrm{F1}} = (40.01, 41.60)$, all indicating significant improvement once again.

Additionally, *Best F1* model displays the lowest standard deviation values, signifying that the model results are more consistent across the folds.

However, the same trend of diminishing returns of augmentations for our models can be observed as for 4 classes. In this case, Cheng et al. [5] model gains approximately 5% increase, and Cavanagh et al. [4] model gains 4% increase in average metric values, while our model performance increases by a similar amount of $1{-}1.5\%$, same as in the 4-class experiment. In the case of both Top 10 models, average F1 score almost did not improve at all. Though once again, our *Best Loss* model operates similarly to other models. Even though Cavanagh et al. [4] model outperforms our *Best Loss* model with $p_{\mathrm{F1}} = 0.01$, comparison with Cheng et al. [5] yields $p_{\mathrm{F1}} = 1$. This observation further strengthens the idea that our proposed layers help with regularization.

Another important aspect is that the 11-class experiment is based on Barchi et al. [2] experiment, and therefore the results are comparable. However, full comparison of all metrics is slightly difficult, as they have only reported a single accuracy score, and did not perform cross-validation. Nonetheless, when considering their results for $K \geqslant 5$ and $K \geqslant 10$, where the reported accuracy is 63.0%, our proposed architectures outperform them even without the need of augmentations. As authors did not present results for multiple folds, p-value cannot be calculated properly. Nonetheless, the results are slightly better even within the range of 1σ.

As for the best results of $K \geqslant 20$, which represent the brightest galaxies, the reported result of 65.2% [2] is outperformed by using Cavanagh et al. [4] architecture together with augmentations.

However, it is important to mention that the data used in this study for the 11-class experiment is closest to $K \geqslant 5$, although despite that, we are able to achieve better results on the entire GZ2 dataset compared to the best results of Barchi et al. [2] on the brightest galaxies ($K \geqslant 20$).

Table 4. Cross-dataset training results for *Top 10 F1* models.

Augmentations	F1 (σ)	Accuracy (σ)	Precision (σ)	Recall (σ)
	(a) 4-class model applied on 11 classes			
No augmentations	86.85 (0.31)	86.93 (0.33)	87.68 (0.76)	86.20 (0.43)
Augmented	88.61 (0.46)	88.48 (0.47)	89.27 (0.38)	88.07 (0.56)
	(b) 11-class model applied on 4 classes			
No augmentations	44.90 (0.74)	63.53 (0.35)	51.48 (2.54)	43.50 (0.78)
Augmented	44.86 (1.09)	63.90 (0.99)	49.19 (1.65)	43.44 (1.10)

5.4 Cross-dataset performance

In Sections 5.2 and 5.3, our models showed great results compared to other authors' models, though the models were tuned for the specific dataset. Hence, it is important to evaluate whether the models trained for the 4-class task are suitable for the 11-class task, and vice versa.

To do that, *Top 10 F1* models were taken, as F1 was the most important metric to optimize due to the data imbalance (see Section 3.1). The models were adjusted so that they produced the opposite number of outputs, without changing any other aspects of the architecture. The training process was exactly the same as described in Section 4.2. The results are shown in Table 4.

While 11-class-specific model performing on 4-class dataset achieves just slightly worse results with $\overline{p_{\mathrm{F1}}} = 0.047$ (hereinafter $\overline{p_{\mathrm{F1}}}$ denotes an average of p-values) compared to all our 4-class-specific models without augmentations, however, the results become similar with $\overline{p_{\mathrm{F1}}} = 0.24$ when augmentations are used. As for 4-class-specific model performing on 11-class dataset, the similarity is very prominent: $\overline{p_{\mathrm{F1}}} = 0.52$ without augmentations and $\overline{p_{\mathrm{F1}}} = 0.50$ with augmentations. These metrics indicate that models tuned for 4-class data perform similarly well compared to models tuned for 11 classes, and vice versa. This allows to conclude that the architectures developed using the observations outlined in Section 4.1.4 are suitable for different tasks of varying difficulty. Nonetheless, tuning is still recommended due to average metric values are slightly lower than our best performing models that were specifically tuned for the task.

5.5 Ensembles

After performing model tuning (see Section 4.1), it can be observed that models for each classification task are similar, yet have slightly different hyperparameter values. For this reason, it is worth checking if an ensemble of *Best Loss*/F1, and *Top 10 Loss*/F1 models could achieve better results.

Ensemble is built using all 4 models. The raw outputs (after softmax) of all models are taken, multiplied by the model coefficient, and added together. The class which has the highest value is chosen as the output class of the ensemble.

Model coefficients were chosen using 10 iterations of the Sequential Least Squares Programming (SLSQP) algorithm to optimize for the best average F1 score across all 10

Table 5. Ensemble model results.

Augmentations	F1 (σ)	Accuracy (σ)	Precision (σ)	Recall (σ)
	(a) 4-class dataset			
No augmentations	88.75 (0.30)	88.65 (0.32)	89.51 (0.58)	88.11 (0.29)
Augmented	89.68 (0.38)	89.44 (0.38)	90.43 (0.41)	89.04 (0.36)
	(b) 11-class dataset			
No augmentations	46.59 (0.53)	65.07 (0.31)	50.87 (2.70)	45.29 (0.56)
Augmented	46.60 (0.53)	65.46 (0.31)	51.40 (2.38)	45.12 (0.55)

folds, with ϵ values ranging from 0.01 to 0.25, with a step of 0.001. Each coefficient was constrained within $(0, 1)$, with the sum of all coefficients needing to be exactly 1. Coefficients were calculated separately for 4- and 11-class models, and separately for models trained without augmentations and with augmentations. The results are displayed in Table 5.

According to Table 5, an ensemble of models without augmentations achieves similar metrics to our augmented models (see Tables 2(b) and 3(b)).

When further inspecting p-values, an ensemble of 4-class models without augmentations achieves similar performance to our augmented models ($\overline{p_{\mathrm{F1}}} = 0.19$), while an ensemble of 11 class models without augmentations is actually better than our augmented models with $\overline{p_{\mathrm{F1}}} = 0.01$.

Furthermore, ensemble of augmented 4-class models actually outperforms augmented Cavanagh et al. [4] 4-class model with $p_{\mathrm{F1}} = 0.01$. This result demonstrates that potentially, instead of performing a costly tuning process, a couple of similar models can be combined into an ensemble to achieve great results. Nonetheless, further testing is needed to verify if any ensemble achieves similar results, though this is outside of the scope of this study.

6 Conclusions

In this study, we introduce two domain-specific layers – Focal and Wave – that are designed to be applied after convolution. After comprehensive evaluation of F1 metrics (due to dataset imbalance, described in Section 3), it can be concluded that:

1. When training without augmentations, models using the proposed Focal convolution layers outperform best results of models used in other studies by a statistically significant amount – Wilcoxon's signed-rank test's p-value for F1 metric $p_{\mathrm{F1}} = 0.002$ was achieved when comparing both *Top 10 Loss* vs. Cavanagh et al. [4] models for 4-class dataset, and *Best Loss* vs. Cavanagh et al. [4] for 11 classes. For more metrics, see Sections 5.2 and 5.3.
2. For both datasets, the proposed layers help with regularization, as even though our models benefit less from augmentations, they still achieve similar results to other augmented models (for example, for 4 classes of our *Best Loss* vs. Cavanagh et al. [4], $p_{\mathrm{F1}} = 0.13$). This means that using the layers might help simplify training process by reducing reliance on augmentations, as there are multiple methods and

their combinations to explore, which might have varying effect on the results. For more details, see Section 5.5.

3. When our models trained without augmentations are combined into an ensemble, they achieve similar (average p-value $\overline{p_{\mathrm{F1}}} = 0.19$ for 4 classes) or even better ($\overline{p_{\mathrm{F1}}} = 0.01$ for 11 classes) results than our corresponding single models trained with augmentations. Additionally, ensemble of augmented 4-class models even achieves better results ($p_{\mathrm{F1}} = 0.01$) than the augmented Cavanagh et al. [4] model. See Section 5.5 for more details.
4. Models tuned for 4-class task perform equally well for 11-class task, and vice versa, with average p-values ranging from $\overline{p_{\mathrm{F1}}} = 0.24$ to $\overline{p_{\mathrm{F1}}} = 0.52$ compared to the performance our models, with only exception being 4-class model applied on 11 class dataset without augmentations being slightly worse with $\overline{p_{\mathrm{F1}}} = 0.047$ (see Section 5.4). This means that the models containing our proposed layers are suitable for different classification tasks, though tuning is still preferable to achieve better results.
5. Based on the tuning results in Section 4.1, the proposed Focal layer is chosen more often than the regular and Wave convolutions, as indicated by *Top 10 Loss* and F1 models for both datasets. Additionally, smaller models (3 convolution layers) achieve better F1 score, while bigger models (4 layers) are achieving the best loss values.

There are opportunities to utilize the idea of domain-specific layers for future improvements as well. First of all, more layers that are based specifically on differences between galaxy types might be beneficial, such as a layer to help detect the presence of a bar. Multiple layers can also be combined for a single input.

Another possibility would be to apply proposed layers on other CNNs or their parts, where the original input sequence is preserved after the convolution, and there is enough data. Additionally, the layers could further be applied to more advanced architectures, such as Convolution Vision Transformers, used by [3].

Finally, we believe that with an upcoming vast amount of data from new surveys, our study will be beneficial for astrophysicists, especially when trying to automate classification of galaxies with similar morphology.

Author contributions. All authors (To.M., A.V.M.M., and Ta.M.) have contributed as follows: methodology, To.M., A.V.M.M., and Ta.M.; formal analysis, To.M., A.V.M.M., and Ta.M.; software, To.M.; validation, To.M., A.V.M.M., and Ta.M.; writing – original draft preparation, To.M.; writing – review & editing, A.V.M.M. and Ta.M. All authors have read and approved the published version of the manuscript.

Conflicts of interest. The authors declare no conflicts of interest.

Acknowledgment. Experiments conducted in this paper were done using Google Cloud and funded by research credits provided by Google LLC. The credits were used to perform model tuning using their Cloud TPU V3-8 resources.

References

1. T.M.C. Abbott, F.B. Abdalla, S. Allam, et al., The dark energy survey: Data release 1, *Astrophys. J. Suppl. Ser.*, **239**(2):18, 2018, https://doi.org/10.3847/1538-4365/aae9f0.
2. P.H. Barchi, R.R. de Carvalho, R.R. Rosa, et al., Machine and Deep Learning applied to galaxy morphology – A comparative study, *Astronomy and Computing*, **30**, 2020, https://doi.org/10.1016/j.ascom.2019.100334.
3. J. Cao, T. Xu, Y. Deng, et al., Galaxy morphology classification based on Convolutional vision Transformer (CvT), *Astron. Astrophys.*, **682**:A42, 2024, https://doi.org/10.1051/0004-6361/202348544.
4. M.K. Cavanagh, K. Bekki, B.A. Groves, Morphological classification of galaxies with deep learning: Comparing 3-way and 4-way CNNs, *Mon. Not. R. Astron. Soc.*, **506**(1):659–676, 2021, ISSN 1365-2966, https://doi.org/10.1093/mnras/stab1552.
5. T. Cheng, C. J. Conselice, A. Aragón-Salamanca, et al., Optimizing automatic morphological classification of galaxies with machine learning and deep learning using Dark Energy Survey imaging, *Mon. Not. R. Astron. Soc.*, **493**(3):4209–4228, 2020, https://doi.org/10.1093/mnras/staa501.
6. S. Dieleman, K.W. Willett, J. Dambre, Rotation-invariant convolutional neural networks for galaxy morphology prediction, *Mon. Not. R. Astron. Soc.*, **450**(2):1441–1459, 2015, https://doi.org/10.1093/mnras/stv632.
7. E.P. Hubble, Extragalactic nebulae, *Astrophys. J.*, **64**:321–369, 1926, https://doi.org/10.1086/143018.
8. R. Katebi, Y. Zhou, R. Chornock, et al., Galaxy morphology prediction using capsule networks, *Mon. Not. R. Astron. Soc.*, **486**(2):1539–1547, 2019, https://doi.org/10.1093/mnras/stz915.
9. N.E. Khalifa, M. Hamed Taha, A.E. Hassanien, et al., Deep Galaxy V2: Robust deep convolutional neural networks for galaxy morphology classifications, in *2018 International Conference on Computing Sciences and Engineering (ICCSE)*, IEEE, Piscataway, NJ, 2018, pp. 1–6, https://doi.org/10.1109/ICCSE1.2018.8374210.
10. D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, preprint, 2014, https://doi.org/10.48550/arXiv.1412.6980.
11. G. Li, T. Xu, L. Li, et al., Galaxy morphology classification using multiscale convolution capsule network, *Mon. Not. R. Astron. Soc.*, **523**(1):488–497, 2023, https://doi.org/10.1093/mnras/stad854.
12. C.J. Lintott, K. Schawinski, A. Slosar, et al., Galaxy Zoo: Morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey, *Mon. Not. R. Astron. Soc.*, **389**(3):1179–1189, 2008, https://doi.org/10.1111/j.1365-2966.2008.13689.x.
13. A. Mittal, A. Soorya, P. Nagrath, D.J. Hermanth, Data augmentation based morphological classification of galaxies using deep convolutional neural network, *Earth Sci. Inf.*, **13**:601–617, 2020, ISSN 1865-0473, https://doi.org/10.1007/s12145-019-00434-8.
14. T. Mūžas, A. V. Misiukas Misiūnas, T. Meškauskas, Large scale study of binary galaxy image classification and the impact of image augmentation techniques, in *Computational Science and Its Applications – ICCSA 2023*, Volume 13957, Springer, Cham, 2023, pp. 402–412, https://doi.org/10.1007/978-3-031-36808-0_27.

15. P.B. Nair, R.G. Abraham, A catalog of detailed visual morphological classifications for 14,034 galaxies in the Sloan Digital Sky Survey, *Astrophys. J. Suppl. Ser.*, **186**(2):427–456, 2010, https://doi.org/10.1088/0067-0049/186/2/427.

16. T. O'Malley, E. Bursztein, J. Long, et al., Kerastuner, 2019, https://github.com/keras-team/keras-tuner.

17. S.S. Shapiro, M.B. Wilk, An analysis of variance test for normality (complete samples), *Biometrika*, **52**(3/4):591–611, 1965, http://www.jstor.org/stable/2333709.

18. B.D. Simmons, C. Lintott, K.W. Willett, et al., Galaxy Zoo: quantitative visual morphological classifications for 48 000 galaxies from CANDELS, *Mon. Not. R. Astron. Soc.*, **464**(4):4420–4447, 2016, https://doi.org/10.1093/mnras/stw2587.

19. C. Szegedy, W. Liu, Y. Jia, et al., Going deeper with convolutions, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ, 2015, pp. 1–9, https://doi.org/10.1109/CVPR.2015.7298594.

20. M. Walmsley, T. Géron, S. Kruk, et al., Galaxy Zoo DESI: Detailed morphology measurements for 8.7M galaxies in the DESI Legacy Imaging Surveys, *Mon. Not. R. Astron. Soc.*, **526**(3):4768–4786, 2023, https://doi.org/10.1093/mnras/stad2919.

21. M. Walmsley, C. Lintott, T. Géron, et al., Galaxy Zoo DECaLS: Detailed visual morphology measurements from volunteers and deep learning for 314 000, *Mon. Not. R. Astron. Soc.*, **509**(3):3966–3988, 2021, https://doi.org/10.1093/mnras/stab2093.

22. S. Wei, W. Lu, W. Dai, et al., Galaxy morphological classification of the legacy surveys with deformable convolutional neural networks, *Astron. J.*, **167**(1):29, 2023, https://doi.org/10.3847/1538-3881/ad10ab.

23. B.L. Welch, The generalization of Student's problem when several different population variances are involved, *Biometrika*, **34**(1–2):28–35, 1947.

24. F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics Bull.*, **1**(6), 1945.

25. K.W. Willett, M.A. Galloway, S.P. Bamford, et al., Galaxy Zoo: morphological classifications for 120 000 galaxies in HST legacy imaging, *Mon. Not. R. Astron. Soc.*, **464**(4):4176–4203, 2016, https://doi.org/10.1093/mnras/stw2568.

26. K.W. Willett, C.J. Lintott, S.P. Bamford, et al., Galaxy Zoo 2: Detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey, *Mon. Not. R. Astron. Soc.*, **435**(4):2835–2860, 2013, https://doi.org/10.1093/mnras/stt1458.

27. X.-P. Zhu, J.-M. Dai, C.-J. Bian, et al., Galaxy morphology classification with deep convolutional neural networks, *Astrophys. Space Sci.*, **364**(4):55, 2019, https://doi.org/10.1007/s10509-019-3540-1.