

Comparative Study of Galaxy morphology classification via Transfer Learning

241040094

Astronomy Unit, Department of Physics and Astronomy, Queen Mary University of London, London E1 4NS, UK

12 December 2025

ABSTRACT

Galaxy morphology classification is a crucial task in astronomy and astrophysics, providing information on galaxy formation and evolution. Traditionally, this classification has been a manual and labor-intensive process requiring significant astronomical expertise. However, advancements in artificial intelligence, particularly deep learning, offer more efficient and accurate solutions. We investigated the application of convolutional neural networks (CNNs) for galaxy morphology classification using the Galaxy10 DECals dataset. In this work, we have compared a custom-built CNN using TensorFlow 2.18 and DenseNet121 and InceptionV3 models pretrained on ImageNet dataset along with various data augmentation and processing techniques to help us better understand what processes help increase performance. Our results indicate that the custom model achieves an accuracy of 40%, while the DenseNet121 and InceptionV3 models achieve 79.6% and 73% accuracy, respectively. DenseNet121 is further improved by preprocessing the dataset by adding image flipping to give 82% accuracy. The superior performance of the pre-trained model underscores the efficacy of transfer learning in astronomical image classification.

Key words: galaxy classification – image processing – convolutional neural networks

1 INTRODUCTION

The Galaxy is the basic celestial building block in the universe, made up of stars, planets, gas, dust and dark matter. Studying the morphology and classification of galaxies is important for understanding the formation and evolution of the universe. As such, accurate and detailed morphological classifications are required to advance research in this area. Traditionally, galaxy classification is a task that is done manually by the scientists in this field. However, given the amount of data that is being generated by modern astronomical instruments, manual classification is not optimal. Also, this type of classification by humans can be subjective and prone to error. At the current stage, with the new Euclid mission that is sending data on 10s of millions of galaxies in a single survey [Banks \(2025\)](#), automatic intelligent classification is the essential to be able to fully utilise this data and any future data from current or planned astronomical missions. Deep learning, as an important branch of artificial intelligence, has shown strong capabilities in image recognition and classification, and has gradually become an important direction of galactic classification research. While considerable progress has already been made using deep learning techniques for this purpose, it is becoming ever more crucial to systematically compare and evaluate existing models. Such comparisons are essential not only for understanding the strengths and limitations of current approaches but also for identifying pathways to further improvement and innovation in automated galaxy classification.

1.1 Related Work

In order to solve these problems, automatic classification methods have been developed rapidly in recent years. These methods rely

primarily on computer vision and machine learning techniques to classify galaxies by analyzing their features. Traditional machine learning methods, such as support vector machines (SVM), random forests, etc., have been successful in some studies, but they often require a lot of feature engineering and have limited performance when dealing with complex morphological structures [Mostafaei et al. \(2024\)](#).

1.1.1 Data distribution

Recent advancements in galaxy morphology classification have improved accuracy and computational efficiency driven by deep learning architectures. Qian conducted a comparative analysis of VGG16, InceptionV3, and ResNet50, demonstrating that InceptionV3 excels across all performance metrics due to its robust inception modules adept at managing galaxy morphology complexities [Qian \(2023\)](#). Wang highlighted the effectiveness of DenseNet201, which achieved an accuracy of 86% on the Galaxy10 DECals dataset, outperforming VGG16 and MobileNetV2. Despite its computational efficiency, the complexity of DenseNet201 resulted in longer training times and challenges in classifying underrepresented galaxy categories [Wang \(2023\)](#). Premanand, Navneeth, et al. (2023) [Premanand et al. \(2023\)](#) evaluated ResNet, Parallel CNN, and VGG16 using the StarGalaxy Classification dataset and additional data from the Sloan Digital Sky Survey (SDSS). They found that Parallel CNN was the most effective achieving an accuracy of 90.08% for classifying between stars and galaxies. Senel employed a hybrid approach using the Galaxy10 SDSS dataset, integrating CNNs with metaheuristic algorithms to optimize neural network parameters to achieve 85% testing accuracy [Senel \(2023\)](#). Kalvankar, Pandit, and Parwate utilized Efficient NetB5 with the Galaxy Zoo 2 dataset and additional SDSS data,

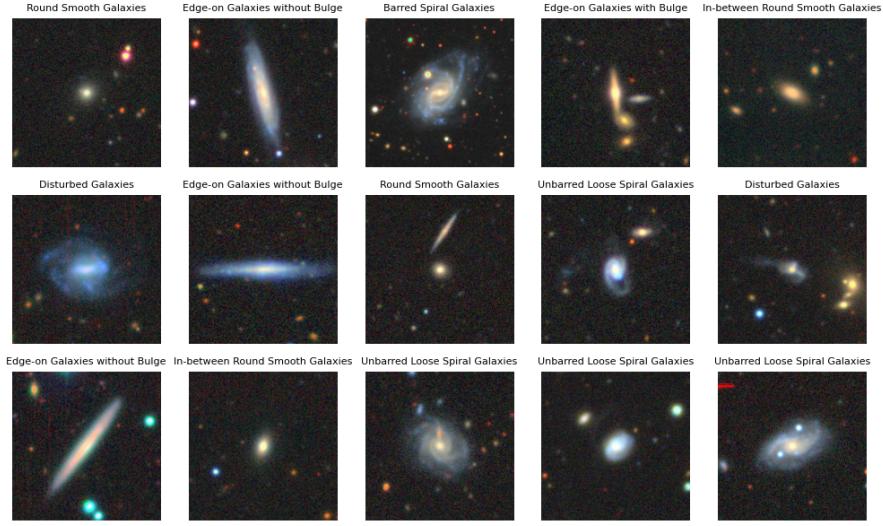


Figure 1. Galaxy 10 Decals dataset sample.

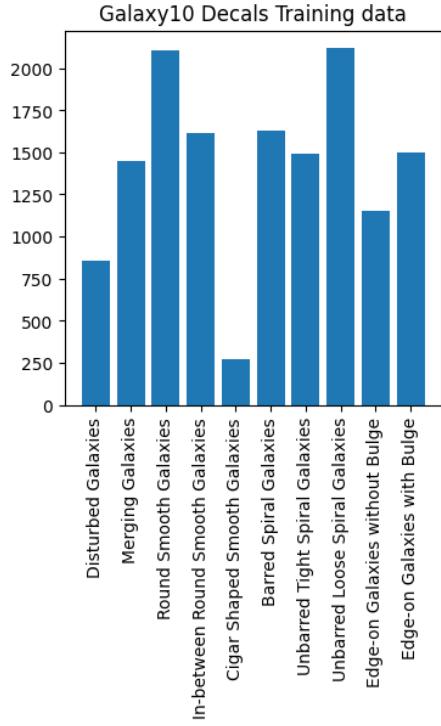


Figure 2. Class label distribution in Galaxy10 DECals dataset

achieving an accuracy of 93.7% and an F1 score of 0.8857 [Kalvankar et al. \(2020\)](#). This study aims to provide a deeper understanding of the data preprocessing steps required for effective galaxy morphology classification and how these steps influence model performance. Specifically, we focus on DenseNet-121 and InceptionV3, identified as one of the best-performing architectures in previous studies [Hui et al. \(2022\)](#); [Fielding et al. \(2021\)](#); [Qian \(2023\)](#). By conducting a series of ablation studies, we systematically evaluate the impact of various data processing techniques on the model's accuracy and robustness. This analysis not only sheds light on the role of preprocessing in deep learning pipelines for astronomical data but also

offers practical insights for optimizing classification models in future work.

1.2 Organisation of report

In section 2 we present the dataset and its distribution. In section 3 we present our methodology and background knowledge required. In section 4 we presented our results, and in section 5 we discuss our findings and provide concluding remarks.

2 DATASET

The Galaxy10 DECals dataset [Leung \(2021\)](#) consists of 17,736 galaxy images from DECals (taken by the Dark Energy Camera) combined with their morphological labels from Galaxy Zoo DR2 (a volunteer citizen science project). Figure 1 shows sample galaxy images from the dataset.

- $y^{(i)}$ is the true class index for the i -th sample
- $\hat{y}_{y^{(i)}}^{(i)}$ is the predicted probability for that class
- $w_{y^{(i)}}$ is the class weight for that true label

The Galaxy10 DECals dataset has data in 10 classes each representing a different galaxy morphology. A distribution of class labels is presented in 4.

The plot shows that the dataset is imbalanced with the class "Cigar shaped Smooth Galaxies" (label 4) having only 259 samples and the label 7 having the highest number of samples at 2097.

2.1 Data Preprocessing

2.1.1 Class Imbalance Mitigation

We use the standard practise of employing class weights to overcome the problem of an imbalanced dataset. This allows us to specify to the classifier to "pay more attention" to the examples from an under-represented. This is equivalent to changing the output bias of the model. In sparse categorical cross-entropy loss (which we will use to train our models) with class weights, the loss function is defined as

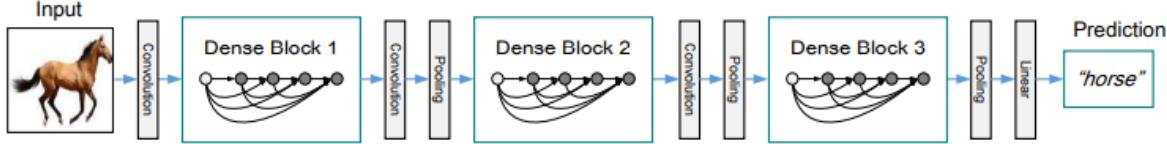


Figure 3. Model Architecture of densenet as presented in Huang et al. (2017)

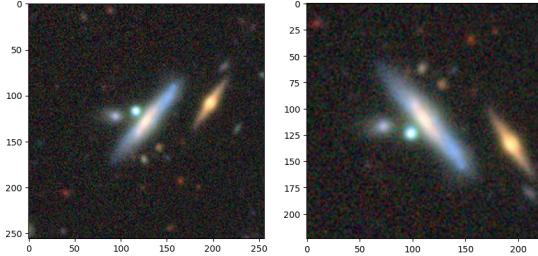


Figure 4. Sample input before and after augmentation

$$\mathcal{L} = -w_y \cdot \log(\hat{y}_y) \quad (1)$$

Where:

- w_y is the class weight for the true class label y
- \hat{y}_y is the predicted probability for the true class y
- The model outputs a probability distribution over C classes

Alternatively, for the full dataset with N samples, the average loss becomes:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N w_{y^{(i)}} \cdot \log(\hat{y}_{y^{(i)}}) \quad (2)$$

Where:

2.1.2 Data augmentation and Normalization

We standardized the image data by scaling pixel values from 255 to 1. This promotes faster convergence during training and improves accuracy. In addition to this, in our experiments we also tried grayscaling and data augmentation.

Grayscaling was used to remove any noise from images related to colour. In the context of galaxy morphology classification, the structural and spatial features of galaxies—such as spiral arms, bulges, and ellipticity—are often more critical than their color characteristics. Grayscaling also helps reduce the number of input color channels(RGB) from 3 to 1, reducing computational complexity and memory required.

In addition, we experimented with data augmentation methods on model performance, including resizing, central cropping and flipping horizontally and vertically. From figure 1, we can also see that for most of the images, the galaxy is only at the center of the image with a lot of negative space which is not useful for training. To counter this, we resized the images to 300x300 and then center cropped it before adding rotation to it. The images were finally resized to 224x224 before being fed into the model. This helps in making our model

Table 1. Baseline CNN Model Architecture

Layer	Type	Parameters	Activation
1	Input Layer	Input shape: (256, 256, 3)	–
2	Convolutional Layer	32 filters, 3x3 kernel	ReLU
3	MaxPooling Layer	2x2 pool size	–
4	Convolutional Layer	64 filters, 3x3 kernel	ReLU
5	MaxPooling Layer	2x2 pool size	–
6	Flatten Layer	–	–
7	Dense Layer	128 units	ReLU
8	Output Dense Layer	10 units	Softmax

more robust by increasing the variety of images for a class type. In our case, this will help the model to not train on a wrong signal - for example orientation of galaxy or spiral turn direction, both of which are not essential for classing a galaxy morphology.

3 METHODS

3.1 Hardware

To accelerate training, we utilized high-performance hardware configurations: We use an NVIDIA A100 GPU, featuring 83.5 GB of RAM and 40 GB of GPU memory. Training times were approximately 400seconds per epoch for nearly all model configurations.

3.2 Baseline CNN Model

We create a custom baseline CNN model to compare the performance of pretrained models built using TensorFlow and Keras layers. 1 shows the model architecture for the baseline model. We keep the same hyperparameters while training this model including number of epochs, batch size and early stopping.

3.3 Transfer Learning

Transfer learning uses knowledge from previously trained models to address similar problems to save computational costs and training time Bengio (2012). It uses the architecture and pre-trained weights of models trained on a large dataset. In our case, we use the DenseNet121 and InceptionV3 models trained on ImageNet. For architecture, we adapted the backbone base model by removing the final layer to better tailor it to the dataset. We then incorporated an global average pooling 2D layer. Then, the output from the adaptive pooling layer was fed into fully connected layers with 1000 and 10

neurons with ReLU and softmax activation functions to get our final output to categorise the classes. We keep the original weights trainable since the images in ImageNet are quite different from the ones in our dataset. This enables our model to utilise the feature extraction properties of the pretrained model and finetune it to our dataset.

3.3.1 DenseNet121

Fig 3 shows the model architecture of Dense

The DenseNet family of models differentiates itself from traditional CNN's by connecting each layer to every other layer in a feed-forward fashion. This works to reduce the number of parameters and alleviate the vanishing gradient problem. DenseNets achieve high performance while requiring less memory and computation when compared to other CNNs. Using DenseNet121 gives us a good balance of computation time when compared with DenseNet201 while maintaining sufficient depth to effectively capture the complex structures present in the dataset

3.3.2 InceptionV3

InceptionV3 is the third architecture developed by Google as part of the Inception family Szegey et al. (2016). It uses asymmetric convolutions of different sizes within the same block, allowing the model to capture features at multiple scales while keeping it computationally efficient by reducing the number of parameters.

4 RESULTS

Before presenting results, we discuss the various metrics we use to judge the performance of our model apart from accuracy. In evaluating the performance of classification models, particularly in cases involving class imbalance, precision and recall are critical metrics. **Precision** is defined as the ratio of true positive predictions to the total number of positive predictions made by the model. It measures the model's ability to correctly identify positive instances without misclassifying negative ones.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

where TP denotes the number of true positives, FP denotes the number of false positives, and FN denotes the number of false negatives.

Recall, on the other hand, is the ratio of true positive predictions to the actual number of positive samples in the dataset. It reflects the model's effectiveness in capturing all relevant instances of the positive class.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

Precision is particularly important in scenarios where false positives carry significant consequences, whereas recall is crucial when false negatives are more detrimental.

Finally, we utilize the confusion matrix to analyze the model's predictions and identify specific areas where its performance is lacking. The confusion matrix provides a detailed breakdown of which classes are being misclassified and into which other categories. This insight enables a more targeted application of data preprocessing techniques and feature extraction strategies to differentiate between

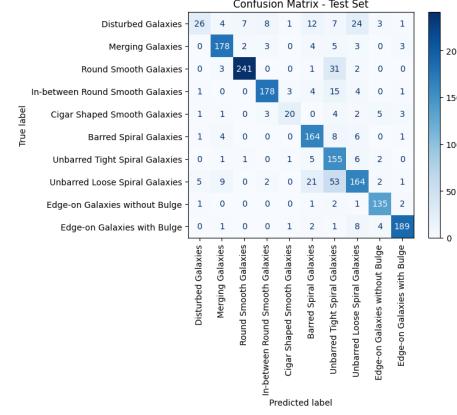


Figure 5. Confusion Matrix

misclassified labels, to guide improvements in the model's overall performance.

Table 2 shows the results of multiple experiments done with DenseNet121 and InceptionV3 and compares it to the BaselineCNN performance:

- Both DenseNet121 and InceptionV3 significantly outperform the Baseline CNN.
- DenseNet121 also outperforms InceptionV3 in this task by a difference of 6 percent in accuracy.
- The best performing model in terms of all metrics is DenseNet121 with image augmentations.

5 and 6 show more metrics on the best performing model. The model trained for 14 epochs and then converged by early stopping. We can see a huge decrease in loss near the 2-3 epoch when the pretrained weights adapt to the dataset and then further improvements as model gets more data. This model took 28 min to train on our hardware 3.1.

From Figure 5, we observe that the model performed reasonably well in classifying Class 4, despite it being highly imbalanced. This could be attributed to the distinct features of Class 4, which likely made it easier for the model to differentiate it from other classes. This observation also helps explain why incorporating class weights into the model resulted in poorer performance compared to the baseline: the adjusted class weights may have negatively impacted training.

Interestingly, the worst-performing class was "Disturbed Galaxies," which had a relatively small number of samples. The model often confused this class with "Unbarred Loose Spiral Galaxies," which had the largest number of samples in the dataset.

When comparing these two classes (see Figure 7), it becomes understandable why this confusion occurred, as they share many similar features.

5 DISCUSSION AND CONCLUSION

• Various CNN architectures including custom-built baseline CNN, training learning with DenseNet121 and InceptionV3 along with various data processing methods were experimented with to explore the potential and limitations of galaxy morphology classification.

- The custom-built baseline CNN achieved a 38.9% overall accuracy.

Model	Weighted Precision	Weighted Recall	Weighted Accuracy	F1Score
Baseline CNN	37.3	38.9	38.9	36.79
DenseNet121	81.1	80	79.6	80
InceptionNetV3	73.3	73.0	73.0	71
DenseNet121 + Augmentation (Rotation)	83	82	82	81
DenseNet121 + Resizing	81.0	80.0	80.0	79
DenseNet121 + Class Weights	78.0	76.0	76.0	76
DenseNet121 + Grayscale	78.0	76.0	76.0	76
DenseNet121 + Class Weights + Resizing	79.0	71.0	71.0	72
DenseNet121 + Augmentation (Rotation) + Resizing	79.0	77.0	77.0	77

Table 2. Performance of multiple models

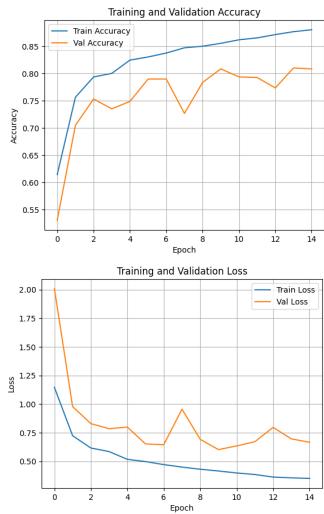


Figure 6. Training And Validation Metrics

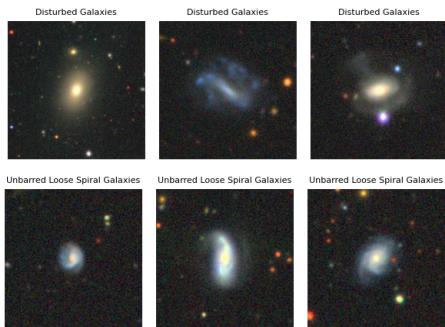


Figure 7. Class 0 and 7 samples

- Pretrained DenseNet121 achieved accuracy of 79.6 and InceptionV3 achieved accuracy of 73 percent. Both were pretrained on ImageNet dataset. Hence, DenseNet121 has superior performance in galaxy morphology classification task

- Augmentation in form of rotation proved to be the most effective in boosting performance. This is possibly due to lack of variety in the

samples in the Galaxy10 DECaLS dataset for each class. Augmentation helps make the model more robust

REFERENCES

- Banks M., 2025, Euclid mission spots 26 million galaxies in first batch of survey data, <https://physicsworld.com/a/euclid-mission-spots-26-million-galaxies-in-first-batch-of-survey>
- Bengio Y., 2012, in Proceedings of ICML workshop on unsupervised and transfer learning, pp 17–36
- Fielding E., Nyirenda C. N., Vaccari M., 2021, in 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), pp 1–5
- Huang G., Liu Z., Van Der Maaten L., Weinberger K. Q., 2017, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
- Hui W., Jia Z. R., Li H., Wang Z., 2022, in Journal of Physics: Conference Series, p. 012009
- Kalvankar S., Pandit H., Parwate P., 2020, arXiv preprint arXiv:2008.13611
- Leung H., 2021, Galaxy10 DECaLS Dataset, <https://astronn.readthedocs.io/en/latest/galaxy10.html#download-galaxy10-decals>
- Mostafaei S. H., Tanha J., Sharafkhaneh A., 2024, Journal of Biomedical Informatics, 157, 104689
- Premananand N., Tarun V., Pawar S., Jawakar D., Deepa S., Jayapriya J., Vinay M., et al., 2023, in 2023 International Conference on Data Science and Network Security (ICDSNS), pp 1–6
- Qian Y., 2023, in Journal of Physics: Conference Series, p. 012009
- Şenel F. A., 2023, Computers, Materials & Continua, 74
- Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z., 2016, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
- Wang G., 2023, in Journal of Physics: Conference Series, p. 012064

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