Improving Galaxy Classification Accuracy Using Multi-Task Learning and Transfer Learning

Lei Yi*

School of Biology and Environmental Science, University College Dublin
Belfield, Dublin 4, D04 V1W8, Ireland

Abstract. In large-scale astronomical surveys, accurate galaxy classification is essential for advancing our understanding of the universe. This study systematically compares and evaluates the performance of three deep learning models for galaxy image classification. This paper uses the Galaxy10 DECals dataset and trained three models: a convolutional neural network (CNN), an autoencoder–classifier joint model and a transfer learning based ResNet50.For three models, this study employed multiple performance-evaluation methods, such as Kullback–Leibler divergence and accuracy, to assess their performance. The results show that adding multi-task learning and transfer learning gradually improves the accuracy of galaxy classification. In particular, the fine-tuned ResNet-50 achieved the highest classification accuracy by leveraging ImageNet pre-trained features via transfer learning. This result underscores the efficacy of transfer learning in bridging the domain gap between natural images and astronomical data. It also provides a robust baseline and practical methodology for employing advanced deep learning models in astrophysical image analysis.

1 Introduction

Galaxy image classification is an important mission in astronomy learning. Its goal is to classify galaxies into different types, which is based on their shape, color and other features. This Process has produced an unprecedented amount of astronomical image data [1]. The large amount of data can not be classified by humans. So powerful automated methods are needed. Deep learning models, especially those like the standard convolutional neural networks (CNNs), have been very successful in image classification. In past studies, it has become a useful tool for tackling galaxy image classification. However, deep learning models still face many challenges in galaxy image classification, such as limited generalization [2], overfitting [3], and how to extract features from galaxy images effectively.

This paper explores and compares the performance of CNN, autoencoder, and ResNet50 on galaxy image classification using the Galaxy10 DECals.h5 dataset. The dataset contains 17,736 labelled galaxy images that span 10 distinct galaxy types. The core mission of this paper is to evaluate the classification performance of different models and analyze their strengths and limitations in improving accuracy and generalization. In addition, this paper

* Corresponding author: lei.yi@ucdconnect.ie

discusses how data augmentation and transfer learning can improve model performance and make it more usable in practice.

To cope with the challenges in automated galaxy classification, such as fuzzy class boundaries, feature uncertainty from observational noise, and limited resolution, and so on. This study designs and tests a progressive iterative method to optimize the models. This method aims to improve classification accuracy and reduce the difference between the model's predicted probability distribution and the true distribution. Its basic idea is to modify only a few controllable factors in each iteration and compare strictly with the previous round. When the improvement is clear and the cost is acceptable, it will be retained; otherwise, it will revert. It ensures that the effect of each change is clear and that similar results can be reproduced under the same settings. The galaxy morphology classification has inherent ambiguity from subjective judgments and fuzzy class boundaries. As a result, this paper uses soft classification with probabilistic labels to reflect this uncertainty [4]. The first optimization objective of this experiment is to reduce the difference between the model's output and this difference is measured in this paper by Kullback–Leibler (KL) divergence. The second point is to improve the model's classification accuracy. The study first built a baseline CNN with four convolutional blocks for feature extraction. After that, the autoencoder was integrated into the framework. It enhances feature learning and helps the model more accurately approximate the true class probability distribution [5]. Finally, the Study added a pretrained ResNet50 to the framework by using transfer learning. This evaluates how powerful ResNet50's deep feature extraction works for this problem and compares it with the other two models.

Experiments show that the CNN gives acceptable performance, the autoencoder improves classification accuracy, and the ResNet50 with transfer learning achieves the best results. Through the progressive enhancement and comparison from CNN to autoencoder to ResNet50, this study offers a tested, effective path to high-accuracy automated galaxy classification. The success of the ResNet50 method shows that transfer learning is an effective strategy for this type of problem. Moreover, the results offer reliable guidance for model selection and performance baselines. It helps handle the massive galaxy datasets from current and future surveys. Those things can equip astrophysics with stronger automated tools and speed up data mining and analysis.

2 Method

To address galaxy image classification, this study compares three deep learning methods. It uses the Galaxy10 DECals.h5 dataset, which contains 17,736 labeled galaxy images and ten classes of galaxy can be seen in figure 1. Before the experiments, the data were preprocessed. Input images were normalized, and labels were converted to one-hot encoding to ensure compatibility with subsequent neural network models [6]. During the experiments, three deep learning models were compared: CNN, an autoencoder–classifier joint model, and ResNet50. For each model, a different optimization strategy was applied during training to make performance better. In order to comprehensively evaluate the performance on galaxy image classification, multiple metrics were used. It includes classification accuracy, cross-entropy loss, and Kullback–Leibler (KL) divergence. Through comparative analysis, the effectiveness and advantages of different deep learning models for galaxy image classification are deeply examined. It provides a valuable reference for research in this field.

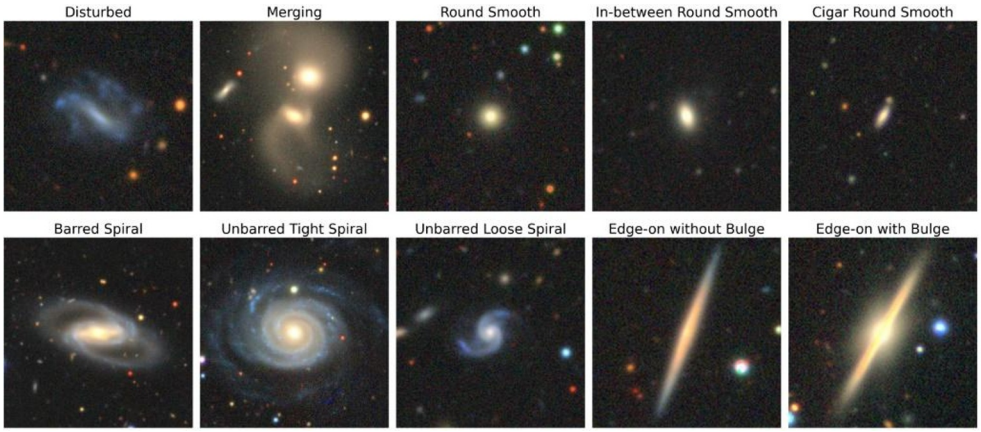


Fig. 1. 10 Galaxy Types from the Galaxy10 DECals Dataset.

2.1 Data Preprocessing

To suit different learning algorithms, the Galaxy10 DECals.h5 dataset was preprocessed before the experiments. Pixel values of input images were converted to 32-bit floating point and normalized to the [0,1] range. This processing has two benefits:

(1) Ensure stability during training. After normalization, the shallow layers near the input are less prone to vanishing or exploding gradients. It stabilizes training and improves efficiency [6].

(2) Standardizing the input data distribution can speed up model training. When the input data distribution changes a lot, training becomes slower and often needs a smaller learning rate and carefully chosen parameter initialization. Normalizing the inputs can reduce those problems [7]. To support subsequent training, labels were converted to one-hot encoding, and label smoothing was used to approximate the true class probability distribution. Finally, as shown in Figure 2, the study uses two-stage stratified sampling to split the dataset into training, validation and test sets. The proportions respectively are 65%, 15% and 20%.

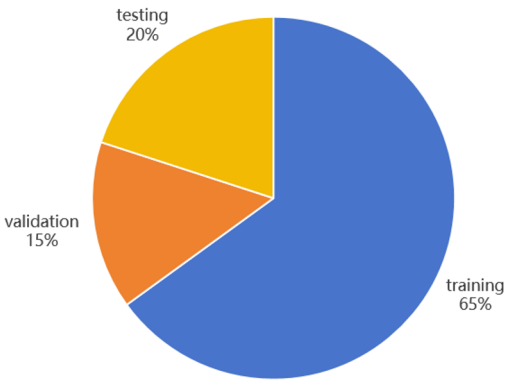


Fig. 2. Dataset split proportions.

2.2 CNN model

CNN is used for image classification. It takes 256×256×3 RGB images as input and outputs a probability distribution over 10 classes. As shown in Figure 3, the network contains four convolutional blocks; each block consists of a 3×3 convolution layer, ReLU activation, batch normalization, 2×2 max pooling and a dropout layer with a rate of 0.25. The network progressively increases the number of feature channels to capture more complex patterns. After a Flatten layer, the feature maps are passed to a fully connected layer with 512 neurons. It also uses ReLU activation and a dropout rate of 0.25 to reduce overfitting. Finally, the class probabilities are obtained with a Softmax layer. The model used the Adam optimizer with a learning rate of 1×10^{-4}, cross-entropy loss, a batch size of 32 and was trained for 50 epochs.

A potential risk of the model is the large number of parameters in the fully connected layer. With 256×256 inputs and four 2×2 pooling steps, the spatial size becomes 16×16; if the last layer has 256 channels, the flattened vector is 16×16×256. A fully connected layer with 512 neurons requires 65,536×512 weights plus 512 biases. It totals 33,554,944 parameters. This size may negatively affect model performance after training [8].

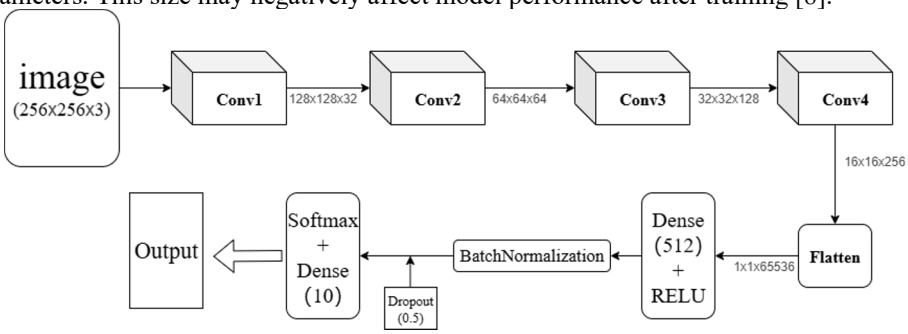


Fig. 3. Architecture of the baseline CNN.

2.3 Autoencoder–classifier joint model

Autoencoder–classifier joint multi-task framework: input preprocessed images and the network outputs a reconstruction and a ten-class probability distribution. As shown in Figure 4, the model uses a shared encoder of four downsampling convolutional layers to compress features. After that, it splits into two branches: One branch is the decoder, which uses transposed convolutions to progressively restore the image and outputs the reconstruction with a sigmoid activation. The reconstruction loss is mean squared error. The other branch is the classifier: the latent vector is passed to a fully connected layer with batch normalization and dropout. It produces a ten-class probability distribution with label smoothing. The classification objective is smoothed categorical cross-entropy. During training, a weighted joint loss is used to combine a pixel-level reconstruction term and a distribution-aligned classification term (implicitly, KL-divergence–based). The emphasis of this stage is on matching probability distributions rather than only maximizing accuracy. An adaptive moment estimation optimizer (Adam) with gradient clipping was used. The learning rate was set to 1×10^{-4}, the batch size was set to 64, and training ran for 70 epochs. After that, the classification accuracy was reported. This optimization strategy combines Adam's adaptive learning rate and robustness to irrelevant features. It effectively improves performance on complex tasks [9]. Gradient clipping effectively prevents exploding gradients and stabilizes training. Proper choices of learning rate and batch size can further improve model performance and generalization [10].

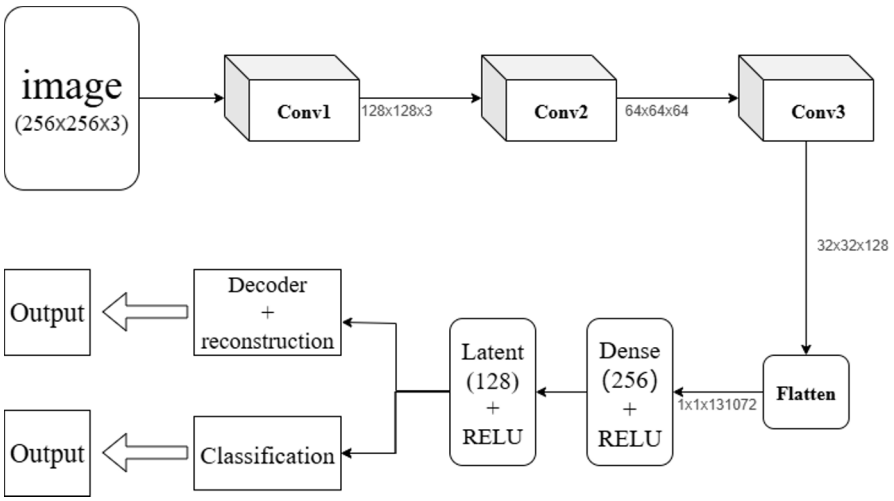


Fig. 4. Architecture of the autoencoder–classifier joint model.

2.4 ResNet50 with transfer learning

ResNet50 is a deep residual convolutional neural network. Its skip connections effectively alleviate the vanishing gradient problem in deep networks. It also endows the model with strong feature extraction capabilities. This study uses a ResNet50 model pre-trained on the large-scale ImageNet dataset. Although the model demonstrates high performance in natural image classification, the ImageNet consists primarily of everyday scenes. Thus, the features identified by the model differ markedly from those of galaxy images. Galaxy images often exhibit faint structures, smooth brightness profiles, and specialized morphological patterns [11]. Consequently, using the pre-trained ResNet50 directly for galaxy classification yields limited model performance. To address this domain gap, transfer learning is adopted. Specifically, the study uses ResNet50's powerful feature extraction backbone while freezing most of the lower-level layers to preserve general-purpose features learned from natural images. After that, unfreeze, fine-tune the top layers, and append custom fully-connected layers tailored for galaxy classification. This approach allows the model to adapt effectively to the astronomical domain while mitigating overfitting and leveraging useful pre-trained representations.

For the structure as shown in Figure 5, weights from an ImageNet-pretrained ResNet50 were loaded, and most lower layers were frozen. The last 20 layers were unfrozen for fine-tuning. It enables the model to retain general features learned from large-scale data while adapting specifically to galaxy data [12]. For the training setting, the Adam optimizer was used with a learning rate of 1e-5. The loss function was categorical cross-entropy with label smoothing. The output layer used Softmax to convert the network's outputs into class probabilities.The core residual unit of ResNet is*:* $\boldsymbol{y = F(x, \{w_i\}) + x}$

The model uses skip connections to alleviate vanishing gradients in deep networks. Evaluation focuses on classification accuracy as the primary metric. Early stopping and adaptive learning rate scheduling were used: training was stopped and the best weights restored if validation performance failed to improve for 10 consecutive epochs; the learning rate was halved if validation loss did not decrease for 3 epochs. This improved training efficiency and generalization.

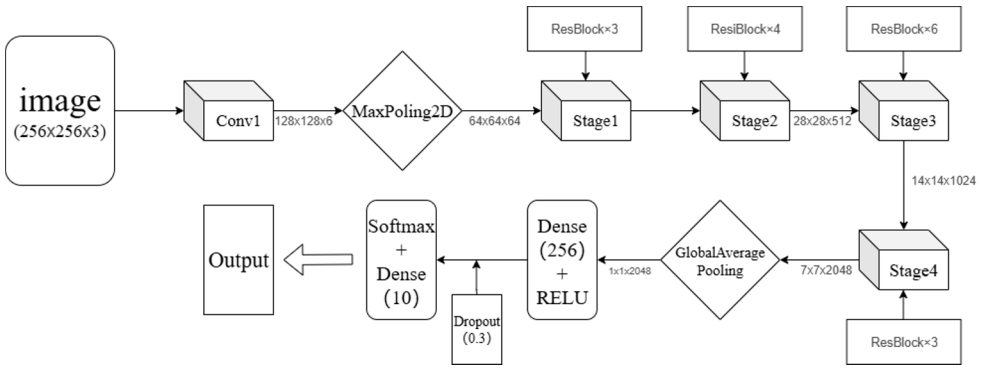
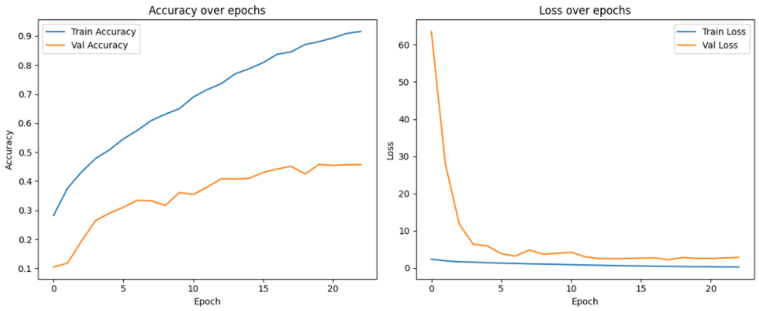


Fig. 5. ResNet50 architecture.

3 Result

3.1 CNN model

As shown in Figure 6, the four-layer CNN reached about 90% training accuracy during training. Its validation accuracy reached about 50%. It indicates severe overfitting. Despite good performance on the training set, the model failed to generalize effectively to unseen data. The model failed to learn broadly transferable features. It results in poor performance on unseen data. This observation aligns with the hypothesis that the fully connected layer has an excessive number of parameters. The large number of parameters increases model complexity and heightens the risk of overfitting. To cope with this problem, an autoencoder-based enhancement was applied: parts of the fully connected layers were replaced with encoder modules to reduce the parameter count. In addition, denoising autoencoders (DAEs) and variational autoencoders (VAEs) were used for data augmentation, generating more useful training samples and easing data scarcity. This approach increased the baseline accuracy from 48.87% to a higher level. This data validates the model's effectiveness in feature learning and model generalization. However, although training accuracy increased and approached 0.9. The validation accuracy improved slowly and stayed around 0.5. It shows that the model is poor in generalization. Training loss dropped quickly and approached zero. It indicates effective learning on the training set. However, validation loss decreased slowly and remained high, further indicating a poor fit on the validation set. Overall, the autoencoder-based enhancement improved performance. The overfitting problem also must be further addressed to strengthen generalization.

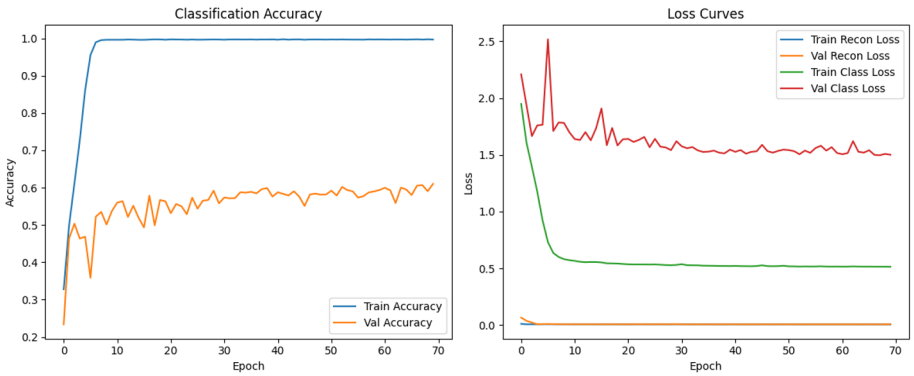


(a) Accuracy of CNN. (b) Loss of CNN.

Fig. 6. Performance of the CNN model on the training and validation sets.

3.2 Autoencoder–classifier joint model

As shown in Figure 7, training accuracy rises from 0.2 to nearly 1.0. However, the validation accuracy increases slowly and stays around 0.7. It indicates limited generalization. The training reconstruction loss dropped quickly to near zero. It shows an effective optimization of the training set. However, the validation reconstruction loss remains high, indicating inadequate reconstruction performance on the validation set. For classification accuracy, the training and validation losses showed large early fluctuations, and it did not converge quickly, further indicating overfitting. The loss curves show a sharp early drop in training loss, with reconstruction and classification losses staying low. Although the validation loss fluctuates at first. After that, it gradually decreases. It indicates that the autoencoder framework effectively improves both reconstruction and classification. In addition, as shown in Figure 8, Data points form clear clusters by class. It further confirms the autoencoder's effectiveness in feature extraction and distribution alignment. The final test results show 61% accuracy on the test set, a KL divergence of 1.0167, and a cross-entropy of 1.5169. These values further highlight the performance gap between the training and test sets. Overall, although overfitting remains in the model. It shows clear advantages in raising classification accuracy, reducing loss, and improving generalization.



(a) Accuracy of Autoencoder. (b) Loss of Autoencoder.

Fig. 7. Performance of the Autoencoder model on the training and validation sets.

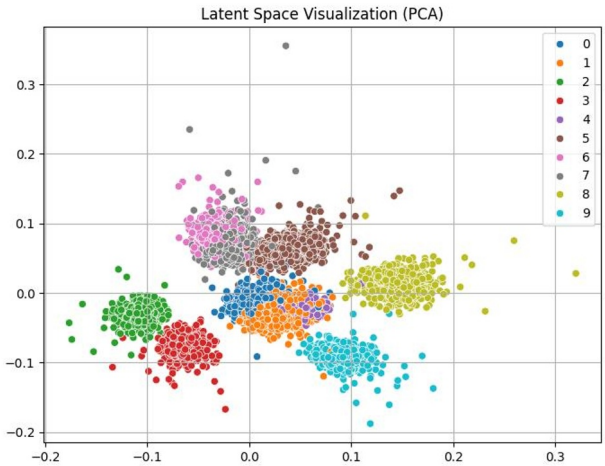
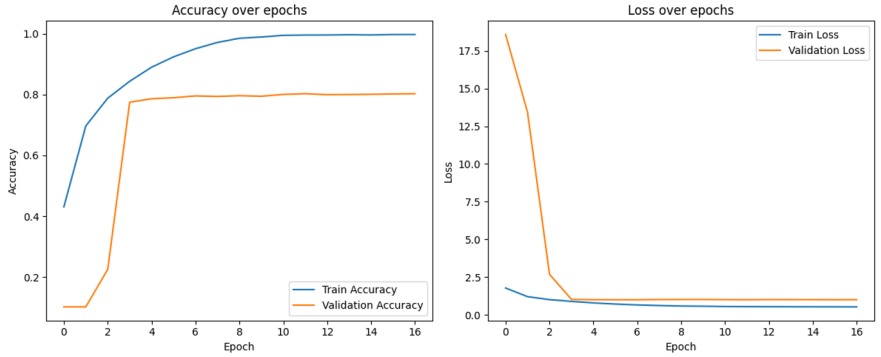


Fig. 8. Two-dimensional PCA visualization of the autoencoder model's latent space.

3.3 ResNet50 with transfer learning

The fine-tuned ResNet50 showed clear improvement during training. As shown in Figure 9, Training accuracy rose quickly and stabilized. Finally, it reached nearly 1.0. Those data indicate the model's effective learning on the training set. However, validation accuracy increased only from about 0.2 to around 0.8. Although a gap remains relative to training accuracy, the validation accuracy improves markedly compared with the unfine-tuned ResNet50. This indicates improved generalization after fine-tuning. The loss curves show that the training loss drops quickly and stays low, while the validation loss decreases markedly. It indicates that overfitting was effectively avoided during optimization. However, validation accuracy grows slowly and plateaus around 0.8. It shows that generalization on the validation set still needs improvement. The final test results show 81% accuracy, a KL divergence of 0.5455 and a cross-entropy of 1.0458 on the test set. It further confirms the model's performance and generalization on unseen data. In conclusion, though the model achieves good results, further data augmentation and regularization are still needed to improve its generalization in practical applications.



(a) Accuracy of ResNet50. (b) Loss of ResNet50.

Fig. 9. Performance of the ResNet50 model on the training and validation sets.

3.4 Discussion

By comparing three deep learning models, this study reveals the key impact of architecture and optimization on galaxy classification performance. The severe overfitting in the CNN shows that complex custom designs on limited data tend to memorize noise rather than learn general features. The very large fully connected layer is the main cause of this issue. The performance gains of the autoencoder–classifier model validate multi-task learning as an effective implicit regularization. The unsupervised reconstruction task pushes the network to learn more general underlying features. Balancing the weights of the reconstruction and classification losses is the key challenge. With transfer learning, ResNet50 delivers excellent performance. It shows that pretrained features transfer strongly in image analysis. Although the domains differ, general features learned on natural images can be fine-tuned for astrophysical tasks. It helps to overcome the scarcity of labeled astronomical data.

During the experiments. Especially when using ResNet50 for transfer learning. It was found that rather than increasing model depth. It is more important to thoroughly analyze the dataset's characteristics before training. And purposefully adjust the pretrained model. For this study, future work will focus on lighter networks(such as ResNet18), more efficient multi-task balancing and testing other advanced models. It can promote the goal of enabling automated astrophysical classification in large-scale surveys.

4 Conclusion

This paper investigates the performance of three models in galaxy classification. The results show that, although CNNs are easy to build, limited generalization makes them prone to overfitting. Introducing a joint autoencoder–classifier multi-task framework strengthens feature learning and modelling of class probability distributions. This yields better performance than the plain CNN. A fine-tuned ResNet50 with transfer learning achieves the best overall results among the compared models. This demonstrates that pretrained models on large datasets can effectively address the specific challenge of galaxy morphology.

The main contribution of the paper is a clear and effective technical path for automated galaxy classification. This path was established through progressive model iteration and comparative experiments. The experimental results confirm the strong potential of transfer learning for this type of task. This paper provides a reliable basis for model selection and a solid methodological foundation for handling large-scale astronomical data in the future.

Despite these advances, research has certain limitations. For example, the performance of all models can be limited by data diversity and quantity. Future work will explore more lightweight network architectures, more advanced regularization technologies, and multimodal data fusion methods to further improve the accuracy, efficiency, and generalization capabilities of models. This could apply them to a wider range of astronomical data mining tasks.

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, 18 (2018)
2. A. Ćiprijanović, D. Kafkes, G. Snyder et al., DeepAdversaries: examining the robustness of deep learning models for galaxy morphology classification. Mach. Learn.: Sci. Technol. 3, 035007 (2022)
3. M. Walmsley, M. Bowles, A. M. M. Scaife et al., Scaling Laws for Galaxy Images. arXiv:2404.02973v1 [cs.CV] (2024)
4. B. Aussel, S. Kruk, M. Walmsley et al., Euclid preparation. XLIII. Measuring detailed galaxy morphologies for Euclid with machine learning. Astron. Astrophys. 689, A274 (2024).
5. D. Doshi, JE. Kim, ReffAKD: Resource-efficient Autoencoder-based Knowledge Distillation. arXiv:2404.09886v1 [cs.LG](2024)
6. E. Fielding, C. N. Nyirenda, M. Vaccari, The Classification of Optical Galaxy Morphology Using Unsupervised Learning Techniques. In: 2022 Int. Conf. Electr. Comput. Energy Technol. (ICECET), Prague, pp. 1–6 (2022)
7. S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Proc. 32nd Int. Conf. Mach. Learn. (ICML), Lille, pp. 448–456 (2015)
8. R. Yamashita, M. Nishio, R. K. G. Do et al., Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018)
9. S. Kalra, J. Wen, J. C. Cresswell et al., Decentralized federated learning through proxy model sharing. Nat. Commun. 14, 2899 (2023)
10. N. Bhatt, N. Bhatt, P. Prajapati et al., A Data-Centric Approach to improve performance of deep learning models. Sci. Rep. 14, 22329 (2024)

11. M. Walmsley, A. M. M. Scaife, C. Lintott et al., Practical galaxy morphology tools from deep supervised representation learning. Mon. Not. R. Astron. Soc. 513, 1581–1599 (2022)
12. C. Gu, M. Lee, Deep Transfer Learning Using Real-World Image Features for Medical Image Classification, with a Case Study on Pneumonia X-ray Images. Bioengineering 11, 406 (2024)