engineering proceedings

MDPI

Proceeding Paper

Enhanced Comparative Analysis of Pretrained and Custom Deep Convolutional Neural Networks for Galaxy Morphology Classification †

Tram Le *, Nickson Ibrahim, Thu Nguyen, Thanyaporn Noiplab, Jungyoon Kim and Deepshikha Bhati

Department of Computer Science, Kent State University, Kent, OH 44242, USA; nibrahim3@kent.edu (N.I.); tnguyen69@kent.edu (T.N.); tnoiplab@kent.edu (T.N.); jkim78@kent.edu (J.K.); dbhati@kent.edu (D.B.)

* Correspondence: hle12@kent.edu

† Presented at the 2024 IEEE 7th International Conference on Knowledge Innovation and Invention, Nagoya, Japan, 16–18 August 2024.

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. We developed and compared three models: a custom-built CNN using TensorFlow 2.18, a ResNet50 model initialized with random weights, and a pre-trained EfficientNetB5 model utilizing transfer learning, both implemented in PyTorch 2.6. Our results indicate that the custom model achieves an accuracy of 67%, while the ResNet50 and EfficientNetB5 models achieve 80 and 87% accuracy, respectively. The superior performance of the pre-trained Efficient-NetB5 model underscores the efficacy of transfer learning in astronomical image classification. These findings have significant implications for the application of deep learning techniques in astrophysical research.

Keywords: galaxy classification; image processing; convolutional neural networks; transfer learning

check for updates

Academic Editors: Teen-Hang Meen, Chun-Yen Chang and Cheng-Fu Yang

Published: 11 March 2025

Citation: Le, T.; Ibrahim, N.; Nguyen, T.; Noiplab, T.; Kim, J.; Bhati, D. Enhanced Comparative Analysis of Pretrained and Custom Deep Convolutional Neural Networks for Galaxy Morphology Classification. *Eng. Proc.* **2025**, *89*, 36. https://doi.org/10.3390/engproc2025089036

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Classifying galaxy morphology remains one of the most challenging tasks in astrophysics. This complexity arises not only from the inherent difficulty in categorizing galaxies based on their images but also from the noise and blurring in these images, which further complicates the classification process. Despite these challenges, recent studies using the Galaxy10 DECals dataset have achieved classification accuracies between 80 and 86%.

Driven by the potential to improve these results, this study aims to increase the accuracy through the application of convolutional neural networks (CNNs). Leveraging advanced machine learning techniques, we have developed a more robust model capable of better handling the intricacies of galaxy morphology. CNNs are particularly well-suited for image classification tasks because they can automatically detect important features without manual extraction, making them ideal for analyzing detailed and varied images of galaxies.

These networks use layers of convolutions that apply filters to input images, creating feature maps that capture essential patterns such as edges, shapes, and textures. For galaxy classification, CNNs can distinguish between the structural features of galaxies.

Furthermore, machine learning techniques such as transfer learning can finetune pretrained networks on new datasets, enhancing learning efficiency and accuracy.

Therefore, in this study, we refine a CNN architecture specifically tailored for classifying galaxies within the Galaxy10 DECals dataset. We leverage transfer learning to boost the performance of our CNN model, resulting in higher accuracy and efficiency. A detailed description of the dataset is provided for preprocessing techniques, model architecture, training process, and evaluation metrics. We conducted a thorough evaluation of our model's performance, including comparisons with existing benchmarks. The findings provide a reference to solve the current challenges in galaxy morphology classification and outline potential directions for future research.

2. Related Work

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. However, the study's focus on a single dataset highlighted potential limitations in its generalizability across various astronomical imaging scenarios [1]. 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 [2]. Premanand, Navneeth, et al. (2023) [3] evaluated ResNet, Parallel CNN, and VGG16 using the Star-Galaxy Classification dataset and additional data from the Sloan Digital Sky Survey (SDSS). They found that Parallel CNN was the most effective, particularly in classifying galaxies and stars, achieving an accuracy of 90.08%. Each model's strengths were identified but a discussion was lacking on the broader applicability and implementation challenges [3].

Senel employed a hybrid approach using the Galaxy10 SDSS dataset, integrating CNNs with metaheuristic algorithms including grey wolf optimizer (GWO), particle swarm optimization (PSO), and artificial bee colony (ABC) to optimize neural network parameters. Although this approach led to a commendable 85% testing accuracy, it showed issues such as sample homogeneity and computational demands affecting diverse galaxy classifications [4]. Kalvankar, Pandit, and Parwate utilized Efficient NetB5 with the Galaxy Zoo 2 dataset and additional SDSS data, achieving impressive results with an accuracy of 93.7% and an F1 score of 0.8857. Despite its strengths, the need for deeper computational algorithm analysis and addressing dataset biases was noted [5].

Such studies deepen the understanding of galaxy morphology classification by highlighting the importance of sophisticated CNN architectures and dataset specificity in achieving high classification accuracy while models such as DenseNet201, InceptionV3, and EfficientNetB5 offer substantial advantages.

3. Dataset

The Galaxy10 DECals dataset [6] 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. Each row represents a class. From top to bottom, their Galaxy10 DECals labels are class 6, 7, and 8.

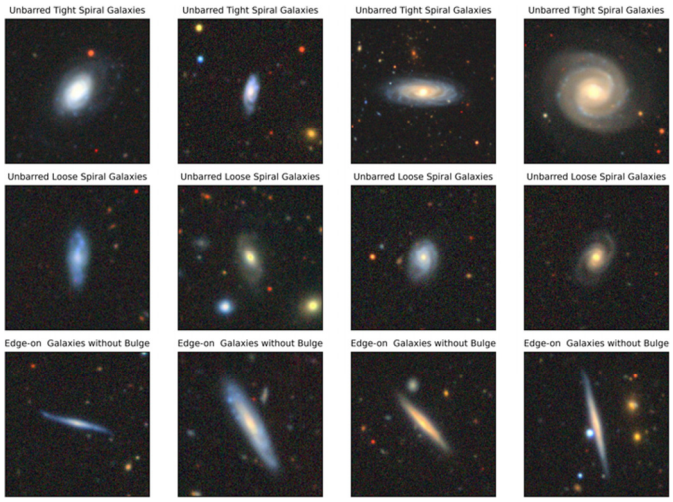


Figure 1. Plotting galaxy images from the dataset. Each row represents a class. From top to bottom, their Galaxy10 DECals labels are class 6, 7, and 8.

3.1. Exploring

The Galaxy10 DECals dataset has better resolution than the original SDSS dataset (Figure 2). Each image is 256 × 256 pixels on 3 spectral bands (g, r, and z) instead of the standard RGB. The images are categorized into 10 classes based on the shapes and features of the galaxies as follows:

- Class 0 (1081 images): Disturbed Galaxies;
- Class 1 (1853 images): Merging Galaxies;
- Class 2 (2645 images): Round Smooth Galaxies;
- Class 3 (2027 images): In-between Round Smooth;
- Class 4 (334 images): Cigar Shaped Smooth Galaxies;
- Class 5 (2043 images): Barred Spiral Galaxies;
- Class 6 (1829 images): Unbarred Tight Spiral Galaxies;
- Class 7 (2628 images): Unbarred Loose Spiral Galaxies;
- Class 8 (1423 images): Edgeon Galaxies without Bulge;
- Class 9 (1873 images): Edgeon Galaxies with Bulge.

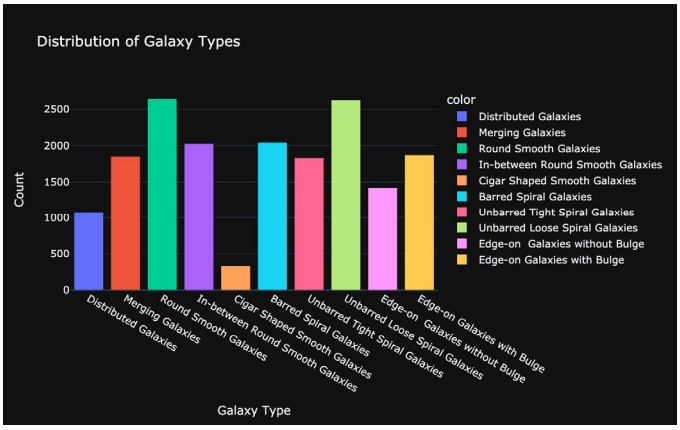


Figure 2. Original distribution of the classes in the Galaxy10 DECals dataset.

3.2. Preprocessing

3.2.1. Class Imbalance Mitigation

We employed data augmentation techniques to address the class imbalance issue, particularly focusing on the underrepresented class of "Cigar Shaped Smooth Galaxies".

By augmenting the existing dataset, we increased the sample size for this class while preserving the original features and characteristics of the images. The augmentation process significantly increased the number of images for the "Cigar Shaped Smooth Galaxies" class. By balancing the classes through augmentation, each class contained comparable samples. We observed a noticeable improvement in model accuracy, with an increase of approximately 24% across all classes.

3.2.2. Data Augmentation and Normalization

We standardized the image data by scaling pixel values from 0255 to 01. Then, we normalized each channel to a mean of 0 and a standard deviation of 1. This approach promotes faster convergence and improves accuracy. In addition, we experimented with different combinations of data augmentation methods on model performance, including rotation, resizing, flipping, shifting, zooming in and out, color jitter, and trivial augment wide. Specifically, we observed an increase in accuracy of approximately 10−12% when utilizing optimized augmentation strategies.

4. Methodology

In this study, we developed and compared three CNN-based models. The first model employs a foundational approach, utilizing convolutional layers to analyze image features. The second model is built with a more complex architecture without relying on pre-trained weights. The third model adopts transfer learning techniques to enhance performance.

4.1. Custom Model

We designed a custom CNN architecture for image processing, incorporating an input layer, four hidden layers, dropout, flatten, and a fully connected layer. The model, depicted in Figure 3, was constructed using the Sequential library, allowing for sequential layer stacking.

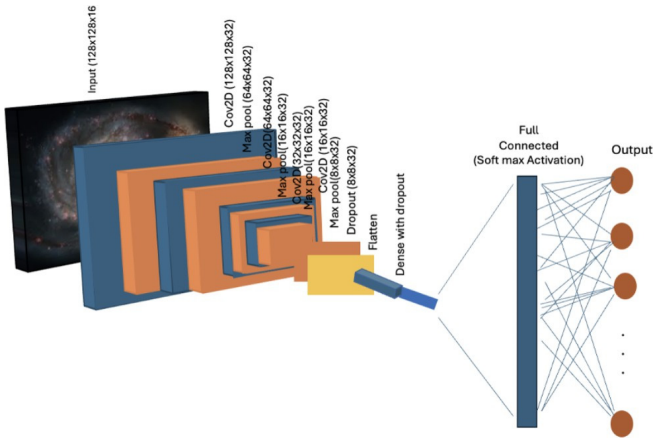


Figure 3. Custom-built CNN model architecture.

The input layer handles 128 × 128, 3-channel color images, initially resized from 256 × 256 images in the Galaxy10 DE Cals dataset. The first convolutional layer employs 16 filters, generating 16 feature maps (128 × 128). Node count increases in hidden layers to enable intricate feature detection. Each convolutional layer is followed by max pooling to downsample inputs while retaining critical features. A 2 × 2 kernel and a stride of 1 pixel in max pooling mitigate noise interference.

4.1.1. Activation Function:

Activation functions play an indispensable role in governing the activation of neurons within the neural network. In this instance, we opted for the Tanh activation function, a choice upheld across all layers except for the output layer. The final layer, responsible for delivering the probability distribution across output classes, leverages the softmax activation function to furnish the requisite probabilities.

4.1.2. Training, Testing, and Validation

Training of the model was orchestrated via the Adaptive Moment Estimation (ADAM) optimizer, a dynamic learning rate optimizer adept at steering the network toward minimal loss values. The training was executed in batches of 32, with each iteration guided by backward propagation and loss computation predicated on categorical cross-entropy. Early stopping was introduced to curtail training sessions upon the discernible absence of loss improvement, warranting a patience threshold of 5 epochs. Impressively, the custom model boasted 98,730 trainable parameters.

4.2. Tailored ResNet50 Based Model

Our model adapted the ResNet50 architecture for the Galaxy10 DECals dataset, leveraging its depth and residual connections for effective training [7]. ResNet50 comprises convolutional and pooling layers, followed by global average pooling and a fully connected layer.

Images were resized from $3 \times 256 \times 256$ to $3 \times 224 \times 224$ before input. We removed the final layer of ResNet50 to align better with our dataset. Following feature extraction, we added an adaptive average pooling layer to preserve spatial information. The flattened output was passed through fully connected layers with 512 and 128 neurons, each followed by ReLU activation to enhance the model's pattern learning ability. A dropout of 0.5 was applied after each layer to prevent overfitting. The architecture includes a fully connected layer for classifying into 10 classes, followed by softmax for a probability distribution. We initialized weights randomly to facilitate training from scratch, allowing the model to learn dataset-specific patterns.

4.3. Transfer Learning with EfficientNetB5 Model

Transfer learning uses knowledge from previously trained models to address similar problems to save computational costs and training time [8]. It uses the architecture and pre-trained weights of models trained on a large dataset, such as ImageNet [9].

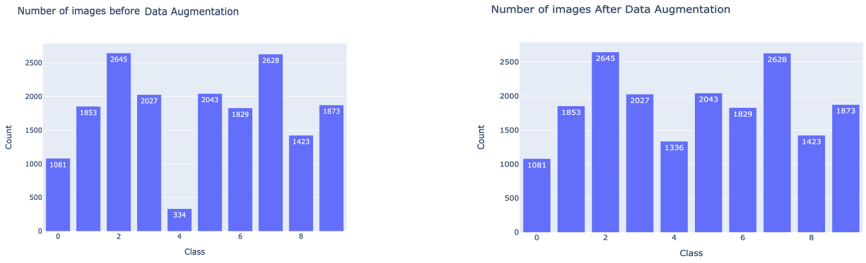
In the method, we froze the early layers of the model so that their weights were not updated during the new training process. For finetuning, we selectively trained the latter layers to adapt the model to new tasks, thereby enhancing its performance on the datasets. This method allowed us to leverage the robust feature-extracting ability of the early layers while customizing the model for targeted applications through fine-tuning.

We experimented with various CNN architectures, including AlexNet [10], DenseNet121 [11], multiple ResNet models, and the EfficientNet [12] series from B1 to B4. EfficientNetB5 performs best in terms of accuracy and computational efficiency. Therefore, we chose EfficientNetB5 and used its pre-trained weights to optimize results without extensive training time. For architecture, we adapted the EfficientNetB5 backbone base model by removing the final layer to better tailor it to the dataset. We then incorporated an adaptive average pooling 2D layer. Then, the output from the adaptive pooling layer was flattened to convert the 2D feature maps into a 1D vector. The dropout rate was 0.5, and subsequent layers included a fully connected layer with 225 neurons and ReLU activation and a final output using softmax to categorize the classes.

5. Results

5.1. Effect of Data Augmentation on Accuracy

Data preprocessing was conducted including class imbalance mitigation and experimentation with data augmentation to increase the model accuracy. By addressing class imbalances and optimizing augmentation strategies, accuracy was enhanced by 15%. The distribution of images for each class before and after balancing is shown in Figure 4.



(**a**) (**b**)

Figure 4. Class distribution of the images before (**a**) and after (**b**) class imbalance mitigation.

5.2. Accuracy and Other Performance Metrics

Our custom model achieved an overall accuracy of 67% as shown in Figure 5.

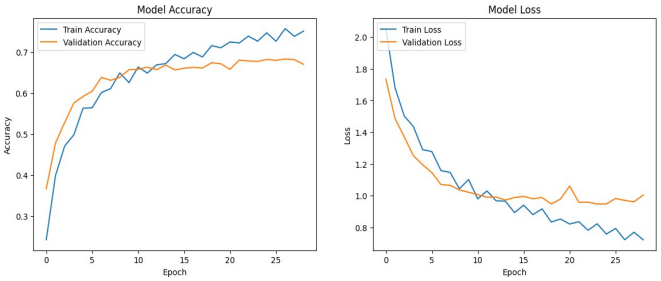


Figure 5. Loss and accuracy curves for training and validation of custom-built CNN model.

As shown in Table 1, the custom model showed an overall accuracy of 67% on 1874 samples distributed across 10 classes. Class 2 showed 83% precision and 84% recall, resulting in an 83% F1 score, while Class 3 presented with a 79% F1 score from 73% precision and 86% recall. In contrast, Class 0 underperformed with only a precision of 36% and a recall of 19%, yielding an F1-score of 25%. Classes 1, 5, 6, and 8 demonstrated moderate effectiveness with F1 scores between 54 and 74%. Figure 6 shows the loss/accuracy curve of the tailored ResNet50 model. The training and validation accuracy curves are closely aligned, with training slightly higher, suggesting minimal overfitting.

As shown in Table 2, the ResNet50 model shows an overall accuracy of 80% across 1774 samples in 10 classes. The model was trained on a dataset split into 90% for training and 10% for testing. The metrics show high precision and recall in most classes, particularly with Class 2 achieving the highest precision at 97%. Class 3 presents the highest recall at 94%. However, Class 0 had notably low recall 19%, indicating the model was difficult to identify this class, possibly due to its underrepresentation or more complex features compared to other classes. Model adjustments improve the detection accuracy for such underperforming classes.

Table 1. Results of performance metrics on multi-class—custom-built model.

Class	Precision	Recall	F1-Score	Support
0	0.36	0.19	0.25	118
1	0.57	0.56	0.56	181
2	0.83	0.84	0.83	268
3	0.73	0.86	0.79	182
4	0.9	0.82	0.86	128
5	0.78	0.56	0.65	231
6	0.66	0.55	0.6	183
7	0.45	0.67	0.54	260
8	0.72	0.76	0.74	139
9	0.74	0.74	0.74	184
Accuracy			0.67	1874
Macro avg	0.68	0.66	0.66	1874
Weighted avg	0.68	0.67	0.67	1874

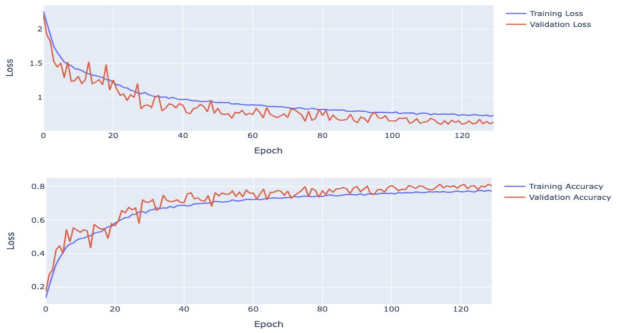


Figure 6. Loss and accuracy curves for train and validation of ResNet50 model.

Table 2. Results of performance metrics on multiclass ResNet50 model.

Class	Precision	Recall	F1-Score	Support
0	0.6	0.19	0.29	111
1	0.83	0.86	0.85	179
2	0.97	0.87	0.92	264
3	0.83	0.94	0.88	213
4	0.68	0.78	0.72	32
5	0.96	0.75	0.84	203
6	0.7	0.83	0.76	193
7	0.62	0.76	0.68	251
8	0.79	0.94	0.86	147
9	0.91	0.86	0.88	181
Accuracy			0.8	1774
Macro avg	0.79	0.78	0.77	1774
Weighted avg	0.81	0.8	0.8	1774

The transfer learning loss/accuracy decreased and stabilized. The training and validation accuracy curves remained close, with validation accuracy occasionally exceeding training accuracy in later epochs, suggesting effective learning of generalizable features without memorization. There was no sign of overfitting, and EfficientNetB5 performed well in both the training and testing phases (Figure 7).

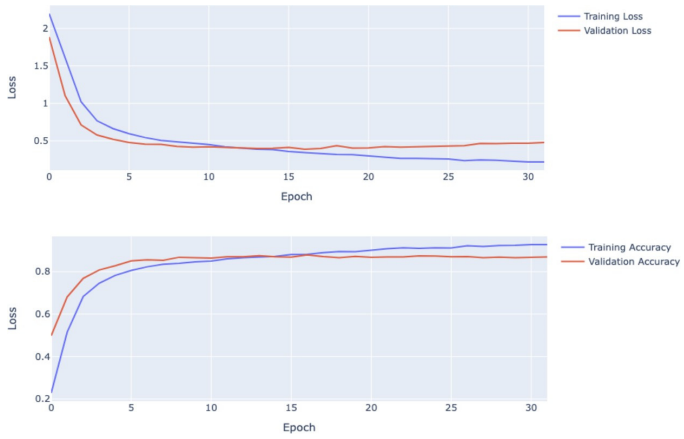


Figure 7. Loss and accuracy curves for train and validation of EfficientNetB5 model.

As shown in Table 3, the EfficientNetB5 model demonstrated strong performance with an 87% accuracy on a dataset of 3548 samples across 10 classes, split into 80% for training and 20% for testing. High precision and recall for Class 2 and 3 were as follows: F1-scores of 94%, a precision of 92%, and a recall of 96%. Class 4, despite having the fewest samples showed an F1 score of 77%. Class 7 presented the lowest F1 score at 78%, indicating variation in class performance. This showcases EfficientNetB5's effectiveness in managing multiclass classification with balanced results across various classes.

Table 3. Results of performance metrics on multiclass EfficientNetB5 model.

Class	Precision	Recall	F1-Score	Support
0	0.71	0.52	0.60	229
1	0.92	0.89	0.90	372
2	0.93	0.96	0.94	546
3	0.92	0.96	0.94	396
4	0.68	0.88	0.77	50
5	0.86	0.90	0.88	377
6	0.80	0.80	0.80	356
7	0.77	0.79	0.78	539
8	0.95	0.93	0.94	309
9	0.95	0.94	0.94	374
Accuracy			0.87	3548
Macro avg	0.85	0.86	0.85	3548
Weighted avg	0.87	0.87	0.87	3548

5.3. Model Performance

The confusion matrix for the first model serves as a discerning evaluation metric, elucidating the model's efficacy across distinct categories. The model's exemplary performance in discerning the "round smooth" category, achieved a commendable 73% precision, 86% recall, and an F1-score of 79%. Conversely, the model exhibited suboptimal performance in categorizing instances of the "Disturbed" category, manifesting a precision of 36%, recall of 19%, and an F1-score of 25%. The second model had diagonal dominance, where predicted labels matched the true labels for most classes, indicating good performance. Figure 8

shows that diagonal dominance was observed for the EfficientNetB5 model, indicating accurate predictions across various classes.

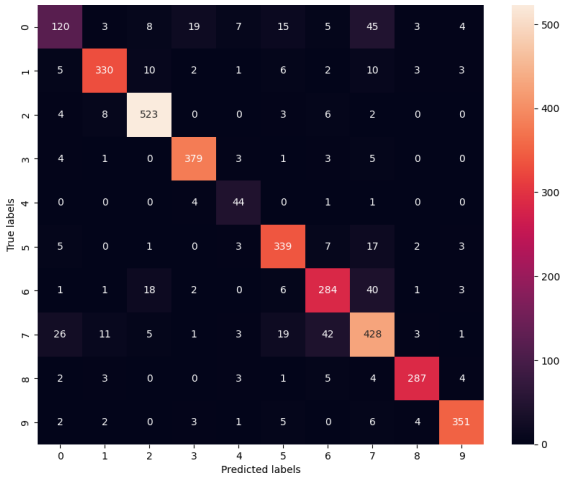


Figure 8. Confusion matrix displaying the classification results EfficientNetB5 model.

5.4. Hardware Specifications

To accelerate training, we utilized high-performance hardware configurations: a Tesla V100 GPU, equipped with 62 GB of RAM and 22.5 GB of GPU memory, with training durations exceeding 20 min. We use an NVIDIA A100 GPU, featuring 83.5 GB of RAM and 40 GB of GPU memory. Training times were approximately 1 h and 30 min for ResNet50 and 25 min for EfficientNetB5.

6. Discussion

Various CNN architectures including custom-built CNN, Tailored ResNet50, and Transfer Learning with EfficientNetB5 were experimented with to explore the potential and limitations of galaxy morphology classification. The custom-built CNN achieved a 67% overall accuracy. To address class imbalances, we employed targeted data augmentation strategies to bolster the representation of underrepresented classes. Augmenting training samples improved accuracy by 24%, notably benefiting previously lower-performing classes. This enhancement underscores the pivotal role of a balanced training set in optimizing deep learning models for diverse galaxy classification tasks. The tailored ResNet50 model showed improved overall accuracy (80%) and demonstrated strong performance across several classes. The consistently high scores in precision and recall for most classes indicate that the model's depth and residual connections effectively capture and leverage the complex features necessary for accurate classification.

The EfficientNetB5 model, utilizing transfer learning, showed the highest accuracy (87%) among the tested models. This superior performance is attributed to the model's ability to effectively generalize from pre-trained ImageNet weights, despite the domain differences between general objects and astronomical images. Notably, this model maintains high precision and recall across almost all classes, demonstrating robustness in handling diverse galaxy morphologies.

Table 4 shows the hyperparameters tuning for three different models: CustomBuilt CNN, Tailored ResNet50, and EfficientNetB5. Architecture and training strategies significantly impact the training duration and computational resources required. The strategic use of data augmentation and class rebalancing is essential in addressing inherent data biases, enhancing model training, and improving classification accuracy.

Table 4. Hyperparameters Tuning for Different Models.

Hyperparameters Tuning	Custom-Built CNN	Tailored ResNet50	EfficientNetB5
Batch Size	32	64	128
Weight	Initialized Randomly	Initialized Randomly	ImageNet
Learning Rate	1.00×10^{-3}	1.00×10^{-4}	1.00×10^{-5}
Optimizer	Adam	Adam	Adam
Loss Function	Cross Entropy	Cross Entropy	Cross Entropy
Momentum	beta_1 = 0.9	betas = (0.9, 0.999)	betas = (0.9, 0.999)
Weight Decay	N/A	1.00×10^{-2}	1.00×10^{-2}
Early Stop Trigger	patience = 5	patience = 15	patience = 10
Epoch	29	120	30
Parameters	98,730	24,624,074	28,804,069

7. Conclusions and Future Work

The significant role of CNNs in classifying galaxy morphology, a task where traditional human classification proves challenging due to the complexity and variability of galaxy images. CNNs, with their ability to autonomously extract and learn features, overcome these challenges, as demonstrated by the varied architectures that we built and analyzed. Addressing the noise in galaxy images is a priority for future work, as reducing noise could further improve model accuracy. Implementing advanced noise reduction techniques and possibly integrating denoising autoencoders in the pre-processing stage could prove beneficial. Furthermore, ongoing efforts to optimize the computational efficiency of our models are essential. Exploring more efficient network architectures or implementing more nuanced training strategies such as federated learning could reduce computational demands and make the training process more feasible on large datasets.

Author Contributions: Conceptualization, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; methodology, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; validation, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; formal analysis, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; investigation, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; resources, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; data curation, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; writing—original draft preparation, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; writing—review and editing, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thu Nguyen) and D.B.; visualization, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab) and D.B.; supervision, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; project administration, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab), J.K. and D.B.; funding acquisition, T.L., N.I., T.N. (Thu Nguyen), T.N. (Thanyaporn Noiplab) and D.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by funds from the Professional Activities Advisory Committee (PAAC) at Kent State University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qian, Y. Performance comparison among vgg16, inceptionv3, and resnet on galaxy morphology classification. *J. Phys. Conf. Ser.* **2023**, *2580*, 012009. [CrossRef]
2. Wang, G. Galaxy morphology classification with densenet. *J. Phys. Conf. Ser.* **2023**, *2580*, 012064. [CrossRef]
3. Premanand, N.; Tarun, V.G.; Pawar, S.; Jawakar, D.N.; Deepa, S.; Jayapriya, J.; Vinay, M. Performance evaluation of convolutional neural networks for stellar image classification: A comparative study. In Proceedings of the 2023 International Conference on Data Science and Network Security (ICDSNS), Tiptur, India, 28–29 July 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
4. Şenel, F.A. A hyperparameter optimization for galaxy classification. *Comput. Mater. Contin.* **2023**, *74*, 4587.
5. Kalvankar, S.; Pandit, H.; Parwate, P. Galaxy morphology classification using efficientnet architectures. *arXiv* **2020**, arXiv:2008.13611.
6. Team, D.S.; Thakar, A.; Malof, J.; Wong, K.; Dey, A.; Schlegel, D. Galaxy10-DECaLS Dataset. 2021. Available online: https://astronn.readthedocs.io/en/latest/galaxy10.html#download-galaxy10-decals (accessed on 1 January 2025).
7. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
8. Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, Bellevue, WA, USA, 2 July 2011; JMLR Workshop and Conference Proceedings; pp. 17–36.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
10. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Van Esesn, B.C.; Awwal, A.A.S.; Asari, V.K. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv* **2018**, arXiv:1803.01164.
11. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
12. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for con volutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.