



# A Lightweight Deep Learning Framework for Galaxy Morphology Classification

Donglin Wu, Jinqi Zhang , Xiangru Li, and Hui Li

School of Computer Science, South China Normal University, Guangzhou 510631, China; [zjq@scnu.edu.cn](mailto:zjq@scnu.edu.cn)

Received 2022 May 4; revised 2022 August 4; accepted 2022 September 7; published 2022 October 19

## Abstract

With the construction of large telescopes and the explosive growth of observed galaxy data, we are facing the problem to improve the data processing efficiency while ensuring the accuracy of galaxy morphology classification. Therefore, this work designed a lightweight deep learning framework, EfficientNet-G3, for galaxy morphology classification. The proposed framework is based on EfficientNet which integrates the Efficient Neural Architecture Search algorithm. Its performance is assessed with the data set from the Galaxy Zoo Challenge Project on Kaggle. Compared with several typical neural networks and deep learning frameworks in galaxy morphology classification, the proposed EfficientNet-G3 model improved the classification accuracy from 95.8% to 96.63% with F1-Score values of 97.1%. Typically, this model uses the least number of parameters, which is about one tenth that of DenseNet161 and one fifth that of ResNet-26, but its accuracy is about one percent higher than them. The proposed EfficientNet-G3 can act as an important reference for fast morphological classification for massive galaxy data in terms of efficiency and accuracy.

*Key words:* methods: data analysis – techniques: image processing – techniques: photometric

## 1. Introduction

Galaxy morphology is an important representation of galaxy evolution, which can be used for exploring their physical properties, reflecting their internal structure and spatial environment. Therefore, classifying the morphology of large numbers of galaxies is considered a long-term goal of astrophysical research (Ellison et al. 2013; Barchi et al. 2020; Hausen & Robertson 2020). However, the volume of galaxy data has exploded due to advances in space observation technologies. The 10 yr Legacy Survey of Space and Time (LSST) program to be carried out by the Rubin Observatory project, for example, will yield 36 TB per night, and generate 500 PB of data totally (Farias et al. 2020). The huge data are far beyond the capacity of manual processing and classification (Abazajian et al. 2009). The only way to deal with the huge amount of data on galaxies is to improve the efficiency of galaxy morphology identification and classification by taking full advantage of the technical advantages in computer science and artificial intelligence.

With the development of artificial intelligence technology, deep learning methods based on convolutional neural networks (CNNs) are currently the most accurate and mainstream research methods (González et al. 2018; Farias et al. 2020). CNN-based methods take images directly as input and use different convolutional kernels and different hidden layers to automatically extract features from images, greatly reducing the level of manual involvement and enabling significant

improvements in classification accuracy (Russakovsky et al. 2015). Cheng et al. (2020) carried out a comparison between several common machine learning methods for galaxy classification, including CNN, k-nearest neighbor (KNN), logistic regression, support vector machine (SVM), random forest and neural networks, and concluded that CNN is the most successful method. Therefore, more and more network architectures based on CNNs were proposed. For example, Zhu et al. (2019) implemented the Residual Networks (ResNets) architecture to classify galaxy morphology on a Kaggle competition data set with an overall accuracy of 95.2083%. González et al. (2018) utilized the DarkNet and real-time object detection system (YOLO) framework for classification of galaxy images. Jiménez et al. (2020) compared an automatically encoded convolutional network with some traditional machine learning classification methods such as KNN, random forests and SVMs, and showed that CNN and ResNet provide better classification results. Farias et al. (2020) built an automatic machine learning pipeline based on the Mask R-CNN network and achieved good results in galaxy binary classification on their data set. Hausen & Robertson (2020) built the Morpheus deep learning framework based on U-Net, demonstrating the high performance of the framework through an application on the CANDELS data set. Cavanagh et al. (2021) developed a novel CNN architecture that outperforms existing models in both 3 and 4-way classification, with overall classification accuracies of 83% and 81% respectively. Tarsitano et al. (2022) extracted features from the galaxy

images by analyzing the elliptical isophotes in their light distribution and collected the information in a sequence, which can accurately classify galaxy images and is faster than other approaches. It can be concluded that various deep learning methods, including ResNets, DarkNet, YOLO, Mask R-CNN, U-Net, etc., have been widely investigated for classification of galaxy morphology and achieved good results.

Although deep learning methods improved the accuracy of galaxy morphology classification, two significant limitations are inevitable. One is that deep learning methods require a large amount of training sample data to learn. According to the study of Miotto et al. (2018), the size of training sample data should be more than ten times the number of parameters in a deep learning model. However, the acquisition of labeled training sample data has always been a challenge for deep learning application. If the number of samples in the training set is not large enough, the heavyweight network is susceptible to the influence from irrelevant information, thus resulting in overfitting. The other limitation is that deep learning needs a large number of computational resources to execute the model effectively for the numerous hidden layers and parameters to tune. As a result, the training and execution efficiency of deep learning methods is relatively low, which requires more than 10 times the computing resources (Barchi et al. 2020). Therefore, how to optimize the structure of deep learning networks to improve the efficiency of galaxy morphology classification is a key issue in performing massive photometric data processing (He et al. 2021).

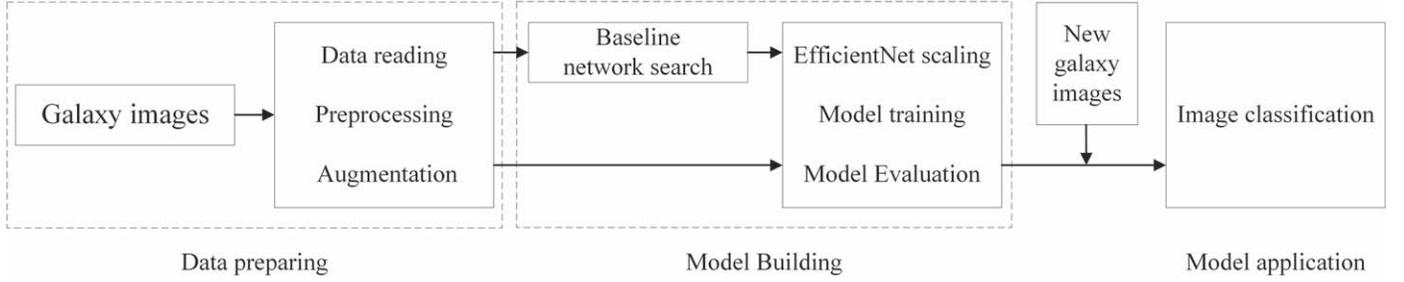
In this scenario, designing a lightweight network becomes a hot topic in deep learning. It has become a major direction in the field of computer vision to build lightweight CNNs by optimizing them in terms of network depth, width and feature resolution. Take EfficientNet-B7 for example, it can process approximately 2 TB of galaxy image data per day by using the NVIDIA RTX 3090 Ti. However, the rapid cadence and scale of the LSST observing program will generate approximately 15 TB per night of raw imaging data (about 20 TB with calibration exposures) and the Square Kilometre Array (SKA) will produce about 5 TB of data per second, and about 2 PB of data per day will be archived (Ivezić et al. 2019). As a result, the heavyweight networks are difficult to adapt to the need for real-time galaxy image classification. With the EfficientNet-G3 designed in this paper, it can process over 300 TB of galaxy image data per day by using the NVIDIA RTX 3090 Ti. This efficiency is more than 150 times that of EfficientNet-B7. This processing speed will also increase with the improvement of computer hardware performance. Obviously, a lightweight network will be more suitable for the rapid growth of astronomical observation data, such as observation data from LSST.

Currently, researchers have constructed several lightweight CNNs for image classification. Iandola et al. (2016) proposed a lightweight and efficient CNN, SqueezeNet, which uses three

main strategies to reduce the number of parameters: replacing  $3 \times 3$  convolutional kernels with  $1 \times 1$  convolutional kernels; reducing the number of input channels for  $3 \times 3$  convolutional kernels; and postponing the down-pooling layer so that the convolutional layer has a larger feature image. SqueezeNet has 50 times fewer parameters than AlexNet, but the classification accuracy is close to that of AlexNet. Howard et al. (2017) introduced deep separable convolution, decomposing standard convolution into depth wise convolution and pointwise convolution, resulting in 8 to 9 times less computation than standard convolution with essentially the same accuracy, and adding a Batch Normalization layer behind the convolution layer to speed up training convergence. MobileNet constructed in this way has 45 times fewer parameters than AlexNet and achieves a classification accuracy of 70.6% on the ImageNet data set. Sandler et al. (2018) introduced the Inverted Residual structure on MobileNet, using  $1 \times 1$  convolution kernels to reduce the number of channels, then  $3 \times 3$  convolution kernels for convolution, and finally  $1 \times 1$  convolution kernels to recover the number of channels and add them to the input, achieving a better accuracy with a smaller number of parameters; at the same time, they introduced the linear bottleneck structure to replace the original non-linear activation transform with a linear transform, and MobileNetV2 was constructed with fewer parameters and higher accuracy than MobileNetV1, achieving 72% accuracy on the ImageNet data set. Howard et al. (2019) built MobileNetV3 on top of MobileNetV2 by replacing some of the  $3 \times 3$  convolutional kernels with  $5 \times 5$  convolutional kernels and introducing the squeeze-and-excite (SE) module and hard swish activation function, and its large version achieved 75.2% classification accuracy on ImageNet. Zhang et al. (2018) utilized channel disruption and group-by-point convolution with depth-separable convolution to modify ResNet to reduce the computational effort of ResNet, reducing FLOPS by 42M compared to MobileNet and achieving 75.3% accuracy on ImageNet. Tan & Le (2019) implemented EfficientNet-B0 as the initial baseline network structure and applied the compound scaling method to amplify it, resulting in EfficientNet-B1 to B7, where EfficientNet-B0 achieved 77.1% accuracy on ImageNet.

According to the above analysis, lightweight networks can improve the computational performance of CNN, which is an effective way to improve the efficiency of galaxy morphology classification. The goal of this paper is to propose a lightweight network for galaxy morphology classification that can take advantage of the accuracy of deep learning and improve the efficiency of galaxy morphology classification, enabling computational support for massive galaxy morphology classification processing tasks.

The paper is organized as follows. In Section 2, we describe the methodology and evaluation method used in this paper. In Section 3, the details of the data sets, data augmentation and neural architecture search (NAS) will be introduced. In



**Figure 1.** Overall framework of lightweight neural networks.

Section 4, we describe the experimental results. The final Section 5 is discussion and conclusion.

## 2. Methodology

### 2.1. Overall Framework

Usually, depth, width and resolution are three important dimensions of a network structure, and previous studies have mostly modified one of these dimensions to achieve higher accuracy, while EfficientNet explores the best combination of the three, achieving a balance between accuracy and efficiency (Tan & Le 2019). The basic idea of EfficientNet includes two steps, the first is to find a baseline network and then scaling the baseline network in depth, width and resolution to find a best network. This paper takes EfficientNet as the basis, and improves the computational speed of the model by modifying the baseline network search method and search space of EfficientNet. The overall process is illustrated in Figure 1.

### 2.2. Baseline Network Search

EfficientNet model construction requires finding the optimal network structure from a large parameter space. To reduce the model search space, the network is scaled only by multiplying the initial baseline network by a constant. Thus, the selection of the baseline network becomes the first task of EfficientNet. In order to improve the search efficiency, this paper integrates the EfficientNet Neural Architecture Search (ENAS) method, which is an efficient neural architecture search method proposed by Pham et al. (2018). It improves the traditional NAS algorithm that requires retraining the entire sub-network model each time when the sub-network is reselected, and allows the previously searched sub-network model parameters to be fully utilized by making all sub-networks share the same copy of model parameters, thus greatly reducing the search time. For real-time processing of massive galaxy images, both morphological classification accuracy and computational power are important factors affecting the morphological classification task, so the objective of the network architecture search is to consider both the accuracy and computational volume, with the

following mathematical expression:

$$\text{Max} \quad \text{ACC}(m) \times \left[ \frac{\text{FLOPs}(m)}{T} \right]^w, \quad (1)$$

where  $m$  stands for a model for evaluation, and  $\text{ACC}(m)$  and  $\text{FLOPs}(m)$  are the accuracy and computational complexity measure of model  $m$ , respectively;  $T$  is the target FLOPs and  $w$  is the hyperparameter used to balance the accuracy and computational effort, with a value of 0.07. The target FLOPs set in this paper is 20M.

In the objective function (1), the accuracy is only one of the multiple performance measures. Apart from the accuracy, actually, the recall, precision and F1 are also often used for evaluating a classification system. Recall is the detection completeness of the interested objects. Precision is the correct ratio of false alarms for our interested objects. In application, we may be interested in replacing the accuracy with the recall in Equation (1) in case that one is especially interested in some specific type of astronomical objects. However, this choice probably results in increased false alarms. Therefore, we can choose the F1 score instead of the accuracy in the objective function (1). The F1 score is the harmonic mean of the recall and precision. More about the recall, precision and F1 score can be found in Zeng et al. (2021).

### 2.3. EfficientNet-based Model Scaling and Construction

Assuming that the entire convolutional network is referred to as  $N$ , its  $i$ th convolutional layer can be thought of as the following functional mapping.

$$Y_i = F_i(X_i), \quad (2)$$

where  $Y_i$  is the output tensor,  $X_i$  is the input tensor and the dimension of the tensor is  $\langle H_i, W_i, C_i \rangle$ , where  $H_i$  and  $W_i$  are the spatial dimensions of the tensor and  $C_i$  is the number of channels of the tensor. Then a convolutional neural network  $N$  consisting of  $k$  convolutional layers can be represented as

$$N = F_k \odot \dots \odot F_2 \odot F_1(X_1) = \odot_{j=1 \dots k} F_j(X_1). \quad (3)$$

In practice, the backbone of a convolutional network is usually divided into multiple structural blocks, each with the

same convolutional layer structure, so that the convolutional network  $N$  can be defined as

$$N = \bigodot_{i=1 \dots s} F_i^{L_i}(X_{\langle H_i, W_i, C_i \rangle}), \quad (4)$$

where the subscript  $i$  denotes the serial number of the structural block and  $F_i^{L_i}$  signifies that the  $i$ th structural block consists of the convolutional layer  $F_i$  repeated  $L_i$  times. In order to reduce the search time, EfficientNet fixes the basic structure of the network, searching only for the network depth ( $L_i$ ), network width ( $C_i$ ) and input resolution size ( $H_i$  and  $W_i$ ), and the network can only be scaled up by multiplying the initial network EfficientNet-G0 by a constant multiplier, so the problem can then be formulated as

$$\begin{aligned} & \max_{d,w,r} \text{Accuracy}(N(d, w, r)) \\ \text{s.t. } & N(d, w, r) = \bigodot_{i=1 \dots s} \hat{F}_i^{d \cdot \hat{L}_i}(X_{\langle r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i \rangle}) \\ & \text{Memory}(N) \leq \text{target\_memory} \\ & \text{Flops}(N) \leq \text{target\_flops}, \end{aligned} \quad (5)$$

where  $w$  is the width,  $d$  is the depth,  $r$  is the resolution multiplier, and  $\hat{F}_i$ ,  $\hat{L}_i$ ,  $\hat{H}_i$ ,  $\hat{W}_i$  and  $\hat{C}_i$  are predefined parameters in the underlying network architecture EfficientNet-G0.

Depth, width and resolution all affect the prediction accuracy of the model. Scaling up any of the three factors can improve the accuracy, but scaling up only one single factor will quickly saturate the accuracy gain, so it is necessary to balance the size of the three factors to improve the accuracy and maintain efficiency. EfficientNet proposes the compound scaling method, which uses a compound coefficient  $\varphi$  to amplify the width, depth and resolution of the network uniformly, based on the following principle.

$$\begin{aligned} \text{depth: } & d = \alpha^\varphi \\ \text{width: } & w = \beta^\varphi \\ \text{resolution: } & r = \gamma^\varphi. \end{aligned} \quad (6)$$

$$\begin{aligned} \text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ & \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{aligned}$$

The mixing coefficients  $\varphi$  are manually adjustable and are used to control the resources consumed due to model expansion.  $\alpha$ ,  $\beta$  and  $\gamma$  are all constants, obtained by neural architecture search, and are used to specify how resources are proportionally allocated to width, depth and resolution.

## 2.4. Evaluation

The models are evaluated by combining accuracy, precision, recall, F1-score and FLOPS required for model. For binary classification, assuming that TP denotes the number of positive samples of correctly classified galaxy patterns, TN corresponds to the number of negative samples of correctly classified galaxy patterns, FP represents the number of positive samples of misclassified galaxy patterns and FN signifies the number of

negative samples of misclassified galaxy patterns, then accuracy, precision and recall are calculated as follows.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \quad (7)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (8)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (9)$$

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (10)$$

The data used in this paper are multi-classified data, while the number of samples in each category varies, so the evaluation of the model relies on the weighted average of the results of each category, and the calculation of the weighted average accuracy, for example, is computed as follows.

$$\text{Weighted Precision} = \sum_i^k W_i \times \text{Precision}_i. \quad (11)$$

$$W_i = \frac{N_i}{\sum_i^k N_i}, \quad (12)$$

where  $k$  denotes the number of galaxy morphology categories;  $W_i$  corresponds to the weight of the  $i$ th category; and  $N_i$  represents the number of samples occupied by the  $i$ th category.

FLOPS can be regarded as a measure of model complexity, and the FLOPS for the convolutional layers in a neural network are calculated as follows.

$$\text{FLOPs} = (2 \times C_i \times K^2 - 1) \times H \times W \times C_0, \quad (13)$$

where  $C_i$  is the number of input feature matrix channels,  $K$  is the number of convolution kernels,  $H$  and  $W$  are the size of the output feature matrix, and  $C_0$  is the number of input feature matrix channels.

The FLOPS for the fully connected layer are calculated as follows.

$$\text{FLOPs} = (2 \times I - 1) \times O, \quad (14)$$

where  $I$  is the number of input neurons and  $O$  is the number of output neurons.

## 3. Experiments

### 3.1. Data Set and Experimental Platform

This paper considers the data set provided by Galaxy Zoo-The Galaxy Challenge competition. This data set contains 61 578 images of galaxies. Based on the Galaxy Zoo clean sample selection method, five types of galaxies are selected for classification, including completely round smooth, in-between smooth, cigar-shaped smooth, edge-on and spiral. Among them, due to the small sample size of cigar-shaped galaxies, in order to increase the sample size of cigar-shaped galaxies, the threshold criterion for cigar-shaped galaxy screening is reduced

**Table 1**  
Clean Samples Selected for Different Classes

Clean sample	Task	Selection	$N_{\text{sample}}$
Completely round smooth	T01	$f_{\text{smooth}} \geq 0.469$	8436
	T07	$f_{\text{completely round}} \geq 0.5$	
In-between smooth	T01	$f_{\text{smooth}} \geq 0.469$	8069
	T07	$f_{\text{in between}} \geq 0.5$	
Cigar-shaped smooth	T01	$f_{\text{smooth}} \geq 0.469$	579
	T07	$f_{\text{cigar-shaped}} \geq 0.5$	
Edge-on	T01	$f_{\text{features/disk}} \geq 0.430$	3903
	T02	$f_{\text{edge-on,yes}} \geq 0.602$	
Spiral	T01	$f_{\text{features/disk}} \geq 0.469$	7806
	T02	$f_{\text{edge-on,no}} \geq 0.715$	
	T04	$f_{\text{spiral,yes}} \geq 0.619$	

in this paper, and the final sample screening rules and sample size are listed in Table 1. The clean sample is divided into a training set and a testing set by sampling each category evenly in a ratio of 9:1.

The model designed in this paper was implemented using Python 3.8.5 in conjunction with PyTorch 1.7.1 and was run using Conda for GPU acceleration, utilizing a computer with 16 GB of RAM and 12 GB of video memory.

### 3.2. Data Enhancement

As the number of images in the training set was less than 30,000, various data enhancement methods were applied in this paper to increase the number of training sets in order to avoid overfitting. The first step of the training set pre-processing is to rotate the images from  $0^\circ$  to  $360^\circ$  and to mirror-flip or horizontally flip the images with a 50% probability. The second step is to crop the images. Since all galaxies in the data set images are located in a small part of the middle of the image and the rest of the image is background unrelated to the galaxy classification task, cropping the central part of the image reduces redundancy in the model input data. In this paper, training size dithering (Simonyan & Zisserman 2014) is used for cropping, which compresses the size of the input image by cropping the image to some random value in a pre-defined range  $S \in [160, 240]$ . In the third step, the brightness, contrast, saturation and hue of the image are modified. In the fourth step, down-pooling the image to the input size  $T$ ,  $T \in [32, 200]$ , is required by the model. In the final step, the image is panned horizontally or up and down by 0–3 pixel points. This method allows the training set to be expanded to 81 times its original size with little or no destruction of the original image information. The training set image pre-processing sequence is depicted in Figure 2. The validation set pre-processing step is divided into two steps: (1) image cropping of the central part to  $200 \times 200 \times 3$ ; (2) image down-pooling to the input size required by the model.

### 3.3. Baseline Network and Mixing Coefficient Search

In this section, the data enhanced by the method described in the above section is used. The model input image size is set to  $32 \times 32$  pixels with Adam as the optimizer and a learning rate of 0.00035. The number of iterations is 300. The network obtained from the search is named EfficientNet-G0, and the structure is shown in Table 2.

Starting with EfficientNet-G0, this paper applies the parameter adaptive configuration method introduced in Section 2.2, which is amplified by the following two steps (Tan & Le 2019).

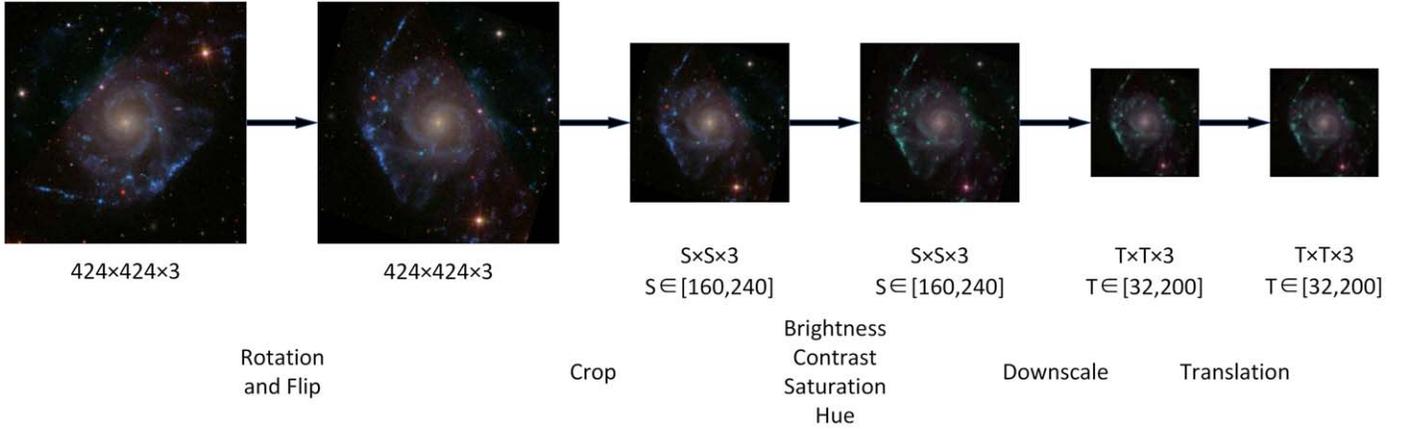
1. STEP 1: we first fix  $\varphi = 1$ , assuming twice more resources available, and do a small grid search of  $\alpha$ ,  $\beta$  and  $\gamma$ , and we find the best values for EfficientNet-G0 are  $\alpha = 1.15$ ,  $\beta = 1.1$  and  $\gamma = 1.2$ .
2. STEP 2: we then fix  $\alpha$ ,  $\beta$  and  $\gamma$  as constants and scale up baseline network with different  $\varphi$ , to obtain EfficientNet-G1 to G3.

## 4. Results

EfficientNet-G3 is the final determined network by the searching method with  $\varphi$ ,  $\alpha$ ,  $\beta$  and  $\gamma$  values of 3, 1.15, 1.1 and 1.2, respectively. Figure 3 displays the confusion matrix of its performance results on the test set, where the horizontal rows are the predicted classification labels and the vertical rows are the actual labels. It can be seen that the type of completely round smooth reached the highest accuracy of 98.46% with the spiral type following. Accuracy of both in-between smooth and edge-on is more than 95%. The type with the worst classification effect is cigar-shaped smooth with an accuracy of only 70.69%. Compared with the work of Zhang et al. (2022), the accuracy of the completely round smooth, in-between smooth and edge-on, is better.

The weighted accuracy, recall and F1-score of different lightweight network models were calculated on the test set, as expressed in Table 3. It can be noticed that EfficientNet-G3 achieves the highest accuracy of 96.6% and the highest F1 value of 97.1% on the test set. EfficientNet-B0 achieved the highest precision rate of 93.7%, but only 95.8% accuracy. According to the comparison results, the proposed EfficientNet-G3 has certain advantages over other models in galaxy morphology classification processing.

To further validate the effectiveness of the model proposed in this paper, EfficientNet-G3 is compared with the methods that have been studied in galaxy morphology classification so far, and Table 4 shows the results of the comparison of the various methods. It can be seen that EfficientNet-G3 designed in this paper requires the least number of parameters, but has the highest accuracy of 96.63% in the classification. Among the networks, DenseNet is the most representative heavyweight network structure proposed by Huang et al. (2017), which


**Figure 2.** Flowchart of data enhancement.

**Table 2**  
Parameters of EfficientNet-G0

Stage $i$	Operator $\hat{F}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv5x5	$32 \times 32$	32	1
2	MBCConv1, k3x3	$16 \times 16$	16	1
3	MBCConv6, k3x3	$16 \times 16$	24	2
4	MBCConv6, k5x5	$8 \times 8$	40	2
5	MBCConv6, k3x3	$4 \times 4$	80	3
6	Conv1x1 & Pooling & FC	$2 \times 2$	160	1

directly connects all layers in the network while ensuring maximum information transfer between the layers connected, and the input of each of its layers comes from the output of all previous layers, i.e., it makes full use of the features of each layer and further alleviates the gradient disappearance problem. However, in terms of classification results, the heavyweight network applied more parameters and the classification accuracy was not necessarily the highest. Kalvankar et al. (2020) directly applied the structure of EfficientNet for galaxy morphology classification, where EfficientNet-B3 achieved 96.35% classification accuracy, second only to our design of EfficientNet-G3. However, EfficientNet-B3 required about six times the number of parameters than EfficientNet-G3 and has a higher latency. Lin et al. (2021) implemented the Vision Transformer (Vit) model, which required the least number of FLOPS and latency, but was only 93.82% in accuracy. Zhu et al. (2019) used ResNet-26 to classify galaxy morphology with an accuracy of 95.21%. Zhang et al. (2022) introduced few-shot learning based on the SC-Net model to classify galaxy morphology with an accuracy of  $94.47\% \pm 0.23\%$ . It can be seen that the other models have much more parameters required, floating point operations and computational latency than the lightweight network we designed; and in terms of accuracy, our trained EfficientNet-G3 has a higher accuracy rate. According to the comparisons, EfficientNet-G3 designed

in this paper has a significant advantage over other commonly used methods.

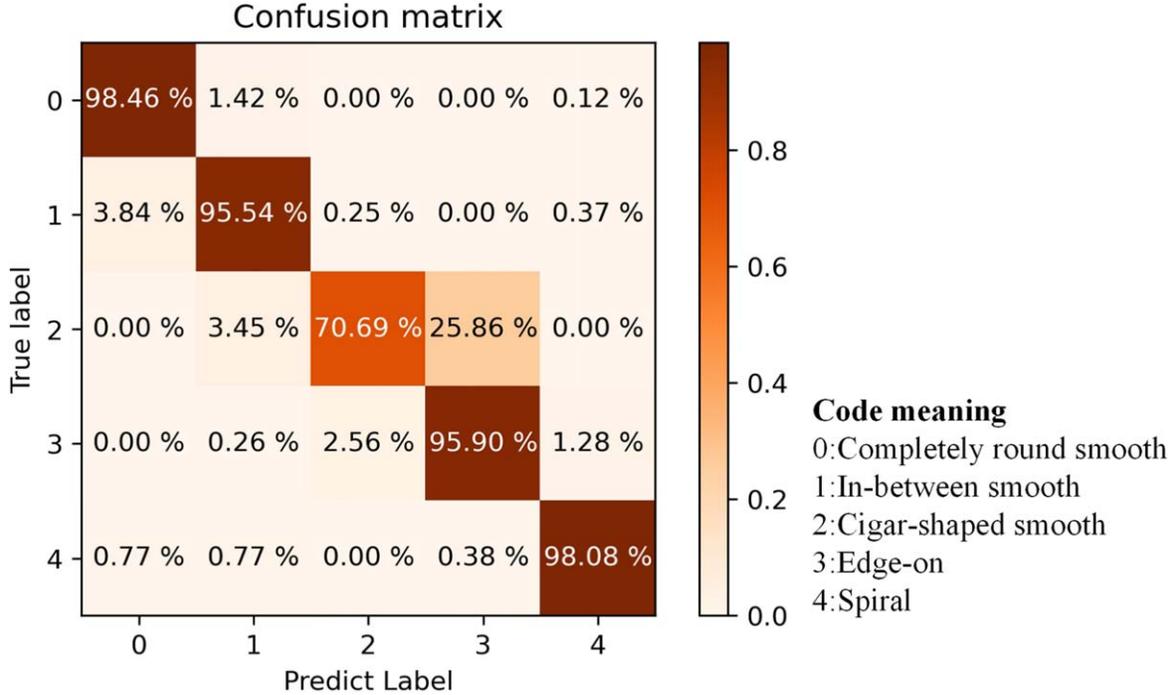
## 5. Discussion and Conclusion

### 5.1. Analysis of FLOPS for Different Networks Models

FLOPS refer to the number of floating-point operations per second, which is often regarded as a measure of the complexity of the model and reflects the model requirements for hardware performance. Figures 4 and 5 show the classification performance of EfficientNet-G0 to G3 compared with other commonly used lightweight frameworks sharing the same parameters with the Adam optimizer, with learning rate set to 0.0001, batch size of 256 and 200 epochs trained.

EfficientNet-G0 is the baseline network obtained by searching on the Galaxy Zoo data set using the ENAS algorithm in this paper, which is 11 times less than EfficientNet-B0 in terms of FLOPS. The deflation factor of EfficientNet-G0 scaled up to EfficientNet-G1 is obtained by applying the compound scaling method with a small grid search on the data set, and EfficientNet-G0 is scaled up with the deflation factor to obtain EfficientNet-G1, G2 and G3, with each version compared to the original EfficientNet-B0 being significantly lower in terms of number of parameters and FLOPS; in particular EfficientNet-G3 can obtain the highest classification accuracy.

The FLOPS of EfficientNet-G2 is 79M and the number of parameters is 0.85M, which is much lower than other commonly used lightweight models, and the accuracy is 95.5%; the accuracy of EfficientNet-G3 is 96.6%, and the closest model to this accuracy is MobileNet-V2 with 96.4% accuracy; there are 1.9 times more FLOPS than EfficientNet-G3, and 1.6 times more parameters than EfficientNet-G3. The overall comparison affirms that EfficientNet-G3 can obtain better classification results with lower number of parameters and FLOPS.



**Figure 3.** Confusion matrix for classification results. (The diagonal lines represent the percentage of correctly predicted data in each category. The remaining values represent the percentage of predictions that were wrong).

**Table 3**  
 Performances of Different Lightweight Deep Learning Models

Model	Accuracy(%)	Precision(%)	Recall(%)	F1(%)
EfficientNet-G0	93.3	90.1	83.3	94.4
EfficientNet-G1	95.0	91.1	86.4	96.3
EfficientNet-G2	95.5	92.2	88.8	96.0
EfficientNet-G3	<b>96.6</b>	92.9	91.7	97.1
EfficientNet-B0 (Tan & Le 2019)	95.8	<b>93.7</b>	89.0	96.3
MobileNet_v2 (Sandler et al. 2018)	96.4	93.5	90.2	96.8
MobileNet_v3_large (Howard et al. 2019)	95.6	91.5	90.1	96.6
MobileNet_v3_small (Howard et al. 2019)	95.2	93.0	89.3	95.5
ShuffleNet_v2_x1_0 (Zhang et al. 2018)	95.3	91.3	88.4	96.4
SqueezeNet1_0 (Iandola et al. 2016)	96.1	93.5	<b>92.7</b>	97.0

*5.2. Performances with Different Resolutions*

Reducing the image input resolution is a common method to reduce the computation of the model, but excessive reduction in resolution can lead to a decrease in model accuracy. In this paper, we compare the accuracy of EfficientNet-G0 after 20 epochs of training at different resolutions, as demonstrated in Figure 6. The experimental results confirm that when the image resolution is less than 32, the accuracy of the model is relatively low, while when the image resolution is greater than 32 and less than 96, increasing the image resolution can effectively improve the accuracy of the model, and then the accuracy starts to decrease. Therefore, in this paper, the

minimum model input image size is set to  $32 \times 32$  and the maximum is set to  $96 \times 96$ .

*5.3. Limitations and Potentials*

The proposed EfficientNet-G3 not only has a smaller number of parameters and FLOPS, but also has better accuracy than the heavyweight networks. However, we should not ignore that the basic net of EfficientNet-G3 is searched on the Galaxy Zoo data set and its hyperparameters are optimized for the galaxy morphology classification task. When using data from other fields, the baseline network and parameters of the model need to be searched again for optimal performance. On the other hand, a

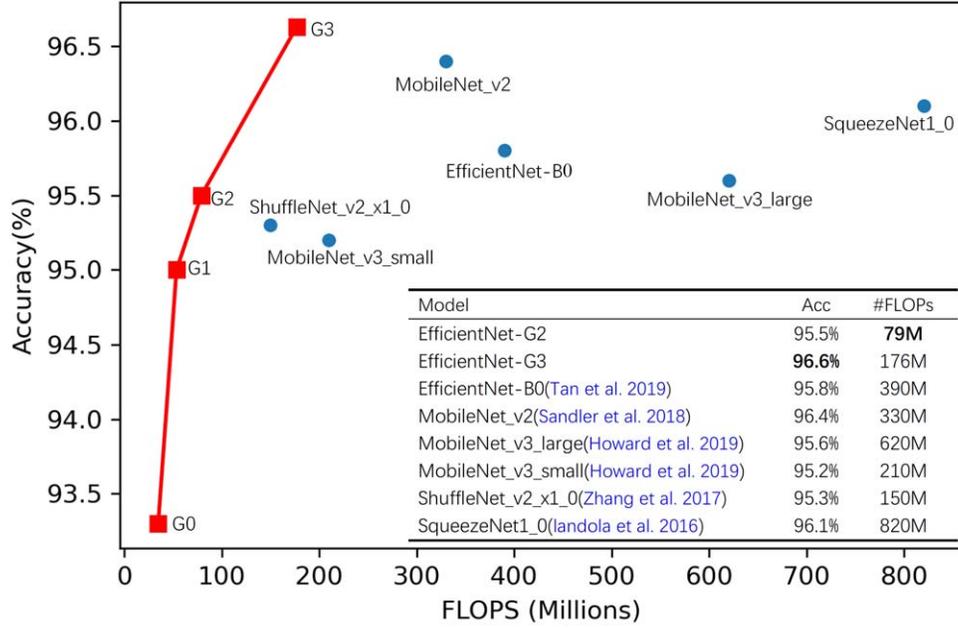


Figure 4. FLOPs and accuracies for different lightweight networks.

Table 4  
Performances of Different Lightweight Deep Learning Models

Model	Params	FLOPS	Acc (%)	Latency (ms)
EfficientNet-G3	2.17M	0.21B	96.63	10.87
DenseNet121 (Huang et al. 2017)	6.96M	2.87B	96.01	16.01
DenseNet161 (Huang et al. 2017)	28.69M	3.40B	95.76	29.27
EfficientNet-B0 (Kalvankar et al. 2020)	5.29M	0.39B	95.49	11.49
EfficientNet-B1 (Kalvankar et al. 2020)	7.80M	0.70B	96.18	14.88
EfficientNet-B2 (Kalvankar et al. 2020)	9.11M	1.00B	96.22	15.18
EfficientNet-B3 (Kalvankar et al. 2020)	12.24M	1.80B	96.35	17.49
EfficientNet-B4 (Kalvankar et al. 2020)	19.35M	4.21B	95.63	21.15
EfficientNet-B5 (Kalvankar et al. 2020)	30.39M	9.88B	96.01	26.07
EfficientNet-B6 (Kalvankar et al. 2020)	43.05M	19.03B	96.04	30.15
EfficientNet-B7 (Kalvankar et al. 2020)	66.35M	37.02B	95.28	36.61
ViT (Lin et al. 2021)	2.47M	0.15B	93.82	8.09
ResNet-26 (Zhu et al. 2019)	11.69M	1.82B	95.21	18.00
Sc-Net (Zhang et al. 2022)	22.85M	3.53B	94.47	20.83

series of researches shows that the architectures and configurations of deep networks have excellent universal properties. Although the parameters need to be re-optimized, we still believe that the proposed model has excellent reference value. As a lightweight network, EfficientNet-G3 is also optimized in terms of network depth, width and feature resolution. This optimization may lose some useful features, so it may be affected by the input data. Galaxy images are characterized by galaxies located right in the center of the image and surrounded by a large amount of information unrelated to galaxy morphology. When the number of samples in the training set is not large enough, the heavyweight network is susceptible to

the influence of irrelevant information, thus causing overfitting, while the lightweight network is better able to avoid overfitting due to the small number of parameters. For example, if the galaxies were not located in the center of the images, the performance of a heavyweight network may deteriorate.

According to the results of our experiments, the lightweight network can better classify Galaxy morphology with higher accuracy and fewer model parameters. These characteristics promote the application of a lightweight network in astronomical data processing. Lightweight networks require fewer computing resources and are easier to be deployed to some lightweight terminal devices, such as mobile phones, which is

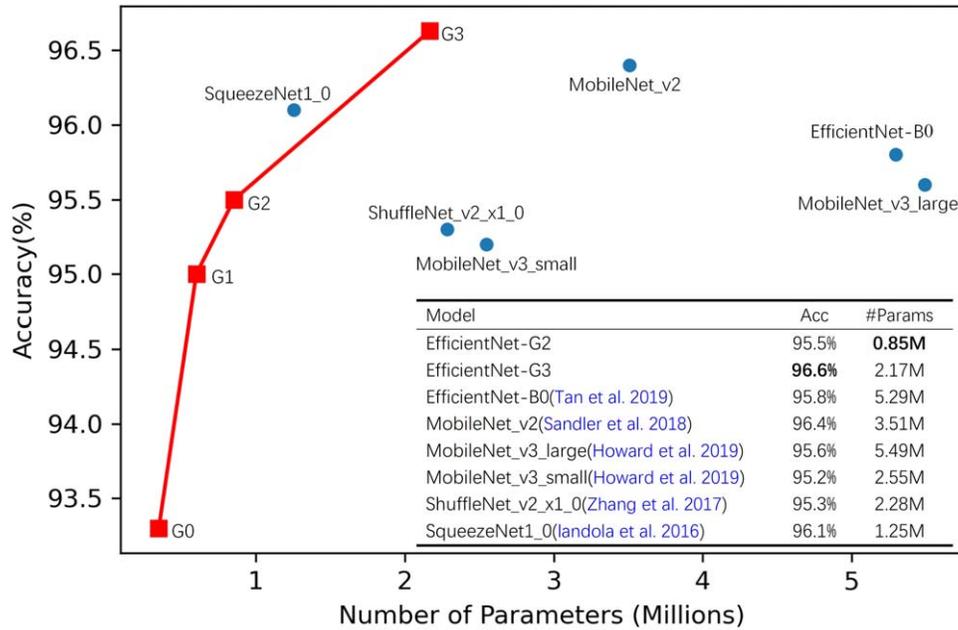


Figure 5. Number of parameters and accuracies for different lightweight networks.

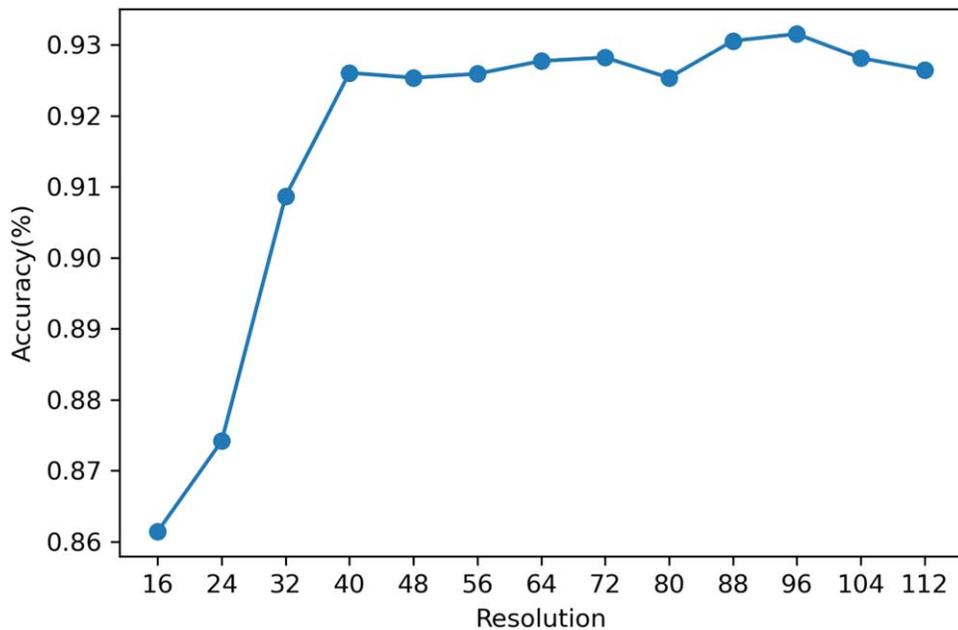


Figure 6. Effect of different resolutions on accuracy.

conducive to promoting public participation and scientific popularization of galaxy morphology classification.

#### 5.4. Future Work

Based on the lightweight network EfficientNet, this paper designs a fast morphological classification framework, EfficientNet-G3, for galaxy images by integrating the ENAS

algorithm. The classification framework first searches the baseline network EfficientNet-G0 using the ENAS algorithm, then applies the compound scaling method to perform a small grid search on the data set to obtain the scaling factor of EfficientNet-G0 to EfficientNet-G1, and finally uses the scaling factor to scale EfficientNet-G0. EfficientNet-G1, G2 and G3 were obtained by scaling up the data set with the scaling

coefficients. Finally, EfficientNet-G3 was identified as the best classification model.

The paper compares the proposed EfficientNet-G3 model with other typical lightweight models and neural networks for galaxy morphology classification, using The Galaxy Challenge data set as an example. The experimental results show that EfficientNet-G3 has significant advantages over other models in terms of the number of parameters, FLOPS and latency. It can adapt to the storage format of massive galaxy data and provide faster processing. However, galaxy morphology classification involves galaxy labeling, data enhancement, training, verification, prediction and other different steps. Next, we plan to combine EfficientNet-G3 with other data processing methods such as semi-supervised or self-supervised to solve the problem of less labeling of galaxy image data sets. We hope we can build an integrated processing pipeline that integrates multiple advantageous methods.

### Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC, Grant Nos. 11973022 and U1811464), and the Natural Science Foundation of Guangdong Province (No. 2020A1515010710).

### ORCID iDs

Jinqu Zhang  <https://orcid.org/0000-0001-6643-4053>

### References

- Abazajian, K. N., Adelman-McCarthy, J. K., Agüeros, M. A., et al. 2009, *ApJS*, **182**, 543  
 Barchi, P. H., de Carvalho, R., Rosa, R. R., et al. 2020, *A&C*, **30**, 100334

- Cavanagh, M. K., Bekki, K., & Groves, B. A. 2021, *MNRAS*, **506**, 659  
 Cheng, T.-Y., Conselice, C. J., Aragón-Salamanca, A., et al. 2020, *MNRAS*, **493**, 4209  
 Ellison, S. L., Mendel, J. T., Scudder, J. M., Patton, D. R., & Palmer, M. J. 2013, *MNRAS*, **430**, 3128  
 Farias, H., Ortiz, D., Damke, G., Arancibia, M. J., & Solar, M. 2020, *A&C*, **33**, 100420  
 González, R. E., Munoz, R. P., & Hernández, C. A. 2018, *A&C*, **25**, 103  
 Hausen, R., & Robertson, B. E. 2020, *ApJS*, **248**, 20  
 He, Z., Qiu, B., Luo, A.-L., et al. 2021, *MNRAS*, **508**, 2039  
 Howard, A., Sandler, M., Chu, G., et al. 2019, in Proc. IEEE/CVF Int. Conf. on Computer Vision (New York: IEEE), 1314  
 Howard, A. G., Zhu, M., Chen, B., et al. 2017, arXiv:1704.04861  
 Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. 2017, in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (New York: IEEE)  
 Iandola, F. N., Han, S., Moskewicz, M. W., et al. 2016, arXiv:1602.07360  
 Jiménez, M., Torres, M. T., John, R., & Triguero, I. 2020, *IEEE Access*, **8**, 47232  
 Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, *ApJ*, **873**, 111  
 Kalvankar, S., Pandit, H., & Parwate, P. 2020, arXiv:2008.13611  
 Lin, J. Y.-Y., Liao, S.-M., Huang, H.-J., Kuo, W.-T., & Ou, O. H.-M. 2021, arXiv:2110.01024  
 Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. 2018, *Briefings Bioinform.*, **19**, 1236  
 Pham, H., Guan, M., Zoph, B., Le, Q., & Dean, J. 2018, in Proc. 35th Int. Conf. on Machine Learning, 80, 4095  
 Russakovsky, O., Deng, J., Su, H., et al. 2015, *Int. J. Comput. Vision*, **115**, 211  
 Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. 2018, in IEEE Conf. on Computer Vision and Pattern Recognition (New York: IEEE), 4510  
 Simonyan, K., & Zisserman, A. 2014, arXiv:1409.1556  
 Tan, M., & Le, Q. 2019, in Proc. 36th Int. Conf. on Machine Learning, 97, 6105  
 Tarsitano, F., Bruderer, C., Schawinski, K., & Hartley, W. 2022, *MNRAS*, **511**, 3330  
 Zeng, Q., Chen, X., Li, X., et al. 2021, *MNRAS*, **500**, 2969  
 Zhang, X., Zhou, X., Lin, M., & Sun, J. 2018, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (New York: IEEE), 6848  
 Zhang, Z., Zou, Z., Li, N., & Chen, Y. 2022, *RAA*, **22**, 055002  
 Zhu, X.-P., Dai, J.-M., Bian, C.-J., et al. 2019, *ApJS*, **364**, 1