

# CLASSIFYING GALAXY IMAGES USING IMPROVED RESIDUAL NETWORKS

Jaykumar Patel<sup>1</sup>, Dan Wu<sup>2</sup>

<sup>1</sup>School of Computer Science, University of Windsor, Ontario, Canada

## ABSTRACT

*The field of astronomy has made tremendous progress in recent years thanks to advancements in technology and the development of sophisticated algorithms. One area of interest for astronomers is the classification of galaxy morphology, which involves categorizing galaxies based on their visual appearance. However, with the sheer number of galaxy images available, it would be a daunting task to manually classify them all. To address this challenge, a novel Residual Neural Network (ResNet) model, called ResNet\_Var, that can classify galaxy images is proposed in this study. Subsets of the Galaxy Zoo 2 dataset are used in this research, one contains over 28,000 images for the five-class classification task, and the other contains over 25,000 images for the seven-class classification task. The overall classification accuracy of the ResNet\_Var model was 95.35% for the five-class classification task and 93.54% for the seven-class classification task.*

## KEYWORDS

*Galaxy Zoo, Deep Learning, Residual Networks, Galaxy Morphology*

## 1. INTRODUCTION

In recent years, astronomy has emerged within the field of natural science, dedicated to the comprehensive study of celestial objects and phenomena [1, 2, 3, 4, 5]. Using mathematics, physics, and chemistry, it seeks to understand the formation and evolution of these objects. Study of astronomy includes studying planets, moons, stars, nebulae, galaxies, comets, and other celestial bodies, as well as phenomena such as supernova explosions, gamma ray bursts, quasars, blazars, pulsars, and cosmic microwave background radiation [1]. Astronomy is the study of anything that originates outside the Earth's atmosphere. With technological advancements over recent decades, astronomy has become a field that generates vast amounts of data [2, 3, 4, 5]. New observational tools such as satellites and telescopes provide large, complex datasets that include spatial and temporal components. One of the primary sources of data used in observational astronomy is photometry, where each image captures a specific field of view of the sky in a chosen frequency band, which can contain multiple objects and is subject to noise [2, 3, 4, 5]. With the simultaneous development of machine learning technologies, it has become possible to handle and extract more value from these massive datasets in various research and industry contexts [4].

An unbiased sample containing reliable morphological types is crucial to any research on extragalactic objects. It allows for accurate classification of galaxies based on their morphologies [3, 4]. Classification of galaxies based on their morphology is vital because the shape and structure of a galaxy can provide insight into its formation and evolution, as well as its interactions with its environment. For example, galaxies in groups or clusters may have different

evolutionary paths than those alone, reflected in their morphologies. To study galaxy formation and evolution, it is essential to classify galaxies into a systematic morphology system.

## 1.1. Galaxy Morphology

Galaxy morphology refers to galaxies' physical makeup and appearance, encompassing traits like size, shape, brightness, and the arrangement of components such as stars, gas, dust, etc. Studying galaxy morphology is crucial in astronomy since it offers a deep understanding of how they form, evolve, and interact with their surroundings. Different kinds of galaxies (spiral, elliptical, or irregular) exhibit unique characteristics that necessitate accurate categorization based on these features so we can understand their properties and behavior. In observational research involving celestial objects, classification into a morphological system is crucial; however, this process can be complex for researchers without a dependable methodology ensuring precise identification. This makes accuracy in classifying celestial objects vital, especially within disciplines where information discovered may reveal valuable insight regarding formation or evolution processes (astronomy).

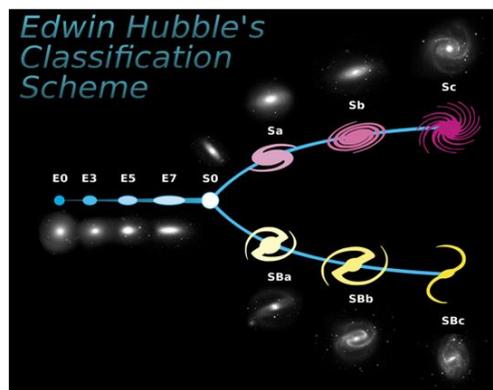


Figure 1. Tuning-fork style diagram of the Hubble sequence (Source: Wikipedia [30]).

Figure 1 shows a classification scheme introduced by Hubble in 1926 [31], which provides a useful starting point for understanding the morphologies of galaxies. Hubble's classification scheme divides galaxies into two main categories, Early Type Galaxies which are characterized by a bulge, giving them elliptical shape and Late Type Galaxies, which are further divided into two categories, spiral galaxies with a bar-shaped central structure and spiral galaxies without a bar-shaped structure at the centre. The classification of galaxy images is an essential task in astronomy, as it enables astronomers to gain insights into the properties and characteristics of galaxies [6]. However, due to the exponential growth of astronomical data, there has been a significant increase in the number of galaxy images requiring classification [7]. Therefore, manual classification is no longer practical, as it is time-consuming, labor-intensive, and prone to human error [8]. One approach to reliable classification is the use of machine learning algorithms. In recent years, there have been significant developments in machine learning, making it a powerful tool for classification tasks in astronomy and other fields. For example, Residual Neural Networks (ResNet) have been successfully used to classify galaxy images based on their morphologies [9]. Other machine learning techniques, such as convolutional neural networks (CNNs) and decision trees, have also been used in astronomical research for classification purposes [10, 11].

## 1.2. Galaxy Zoo

Machine learning algorithms are becoming more common in scientific research, including classifying objects into a morphology system. While machine learning algorithms can significantly improve efficiency and accuracy, human supervision is still necessary to guide the machine and provide meaningful labels to the patterns it discovers. One project that successfully utilized human input is the Galaxy Zoo project [12]. Launched in 2007, this project engaged volunteers in classifying galaxy images into elliptical, spiral, and undefined categories. By combining the power of human intuition and the speed of automated processes, the Galaxy Zoo project could efficiently and accurately classify many galaxy images.

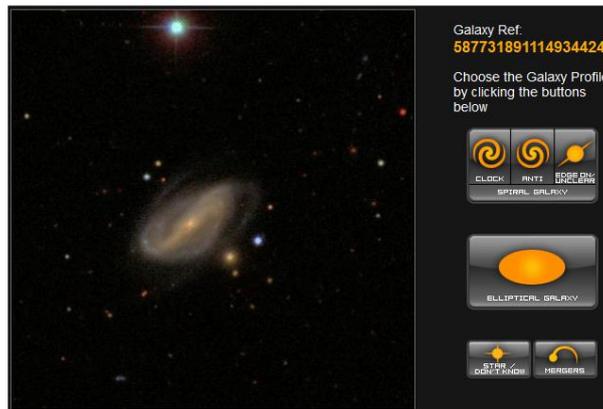


Figure 2. Galaxy Zoo 1 user interface showing a galaxy image on the left and buttons on the right for users to select. By clicking on the buttons, users can choose one of the options from clockwise, anticlockwise, and edge-on, which are under the category of the spiral galaxy, or the elliptical galaxy, or one of the options from star and merger, which are under undefined/ other galaxy category (Source: Lintott et al. (2008) [12]).

Galaxy Zoo is a citizen science project that harnesses the collective intelligence of volunteers to classify galaxies based on their morphologies. The primary objective of GZ1 (Galaxy Zoo 1, 2007) was to categorize galaxies based on their visual appearance, particularly their shapes and structures [12]. The user interface of GZ1 was designed so that users could classify galaxies straightforwardly and intuitively. Figure 2 shows the user interface of Galaxy Zoo 1 [12]. On the left side of the interface, users were shown an image of a galaxy. They were given the option to select the galaxy type from predefined categories such as elliptical, spiral, or other. These three categories are further divided into six options: the elliptical has one option, the spiral has three options listed as clock, anti, and edge-on, and the other is divided into two options for users to select from star and merger. On the right side of the interface, the users were presented with options and examples to help them identify and classify the galaxy type they saw on the left. Users could classify as many galaxies as they wanted and were encouraged to return to the site for more images. The classifications submitted by users were combined and analyzed to create a catalog of galaxy classifications that researchers could use for further analysis [12].

The initial project's success led to the launch of a follow-up project, Galaxy Zoo 2 (GZ2), in 2009. Galaxy Zoo 2 aimed to build on the success of its predecessor by introducing a more complex classification system consisting of a decision tree and several classification stages (Fig. 3). This new system enabled volunteers to provide labeled data for galaxy morphology with unprecedented detail, thus significantly improving our understanding of the universe. The project known as Galaxy Zoo was a significant step towards democratizing scientific research, allowing the public to participate and contribute to discoveries actively. This approach relied on the

collective intelligence of a diverse group of volunteers who could classify and analyze large amounts of astronomical data that would have been unmanageable by a single research team.

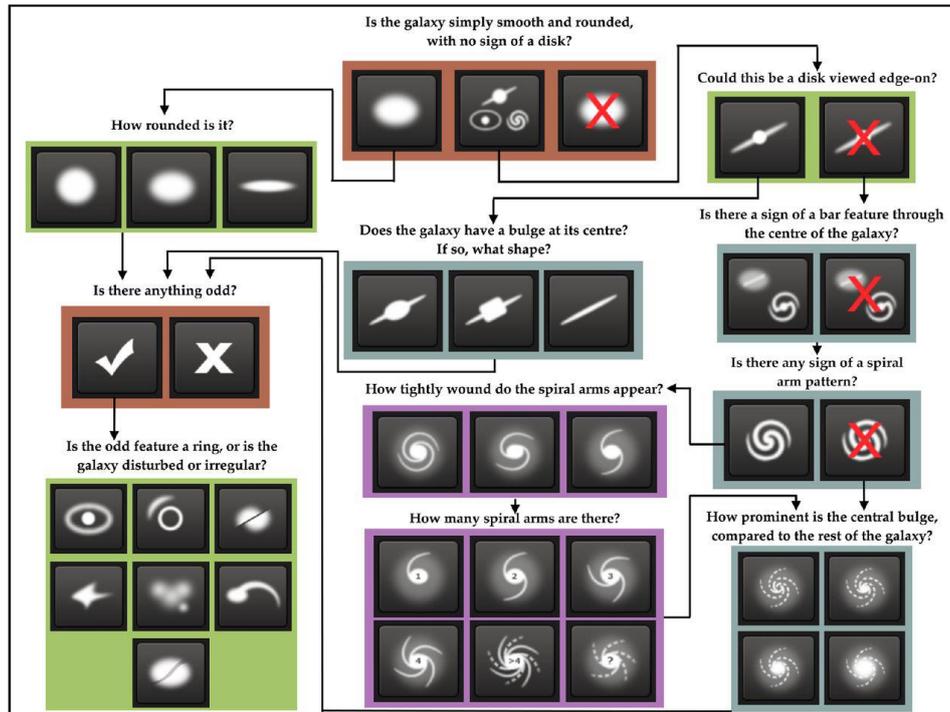


Figure 3. Decision tree used in Galaxy Zoo 2 project (Source: Willett et al. (2013) [29]).

## 2. LITERATURE REVIEW

In recent years, machine learning and deep learning methods have gained popularity in astronomy due to their ability to process large datasets with high accuracy and efficiency. Machine learning and deep learning techniques have been used to classify galaxies based on their shape, size, and color. Using these techniques in astronomy has opened up new avenues for research, allowing scientists to make more accurate predictions and identify previously unknown phenomena. Therefore, it is essential to review the current state of the art in applying machine learning and deep learning techniques to galaxy image classification to understand their potential for advancing the field of astronomy.

The classification of galaxy morphologies has been a challenging task in astronomy, and several approaches utilizing machine learning techniques have been proposed to address this problem. In the paper by Gupta et al. (2022) [13], they present a novel approach for galaxy morphology classification using neural ordinary differential equations. They employ a dataset of galaxy images and train a neural network model based on ordinary differential equations. The results demonstrate the effectiveness of their method in accurately classifying galaxy morphologies. Another paper by Kalvankar et al. (2020) [14] focuses on galaxy morphology classification using efficientnet architectures. They utilize the arXiv dataset and propose a deep learning model based on efficientnet architectures. The experimental results showcase the superior performance of their model compared to existing methods in accurately categorizing galaxy morphologies. Barchi et al. (2020) [15] conducted a comparative study on machine learning and deep learning techniques applied to galaxy morphology classification. They [15] employ various datasets and evaluate multiple models. Their study provides insights into the performance of different approaches, highlighting the strengths and weaknesses of each method. Zhanget al. (2022) [16] explore using

few-shot learning for classifying galaxy morphologies. Their research addresses the challenge of limited labeled data by proposing a few-shot learning framework. By leveraging transfer learning and meta-learning techniques, their approach demonstrates promising results in classifying galaxy morphologies accurately. In the paper by Alawi et al. (2021) [17], the authors propose a deep residual network model for star-galaxy classification. Their model is developed for a specific task within the broader context of galaxy morphology classification. Although the details of the dataset used are not explicitly mentioned in the paper, their model shows promising performance in distinguishing between stars and galaxies. Shetty et al. (2022) [18] tackle the classification of satellite galaxies using convolutional neural networks (CNNs) and machine learning algorithms. Their [18] study focuses on a specific type of galaxy and proposes a CNN-based classification approach. The authors utilize the dataset specific to satellite galaxies and demonstrate the effectiveness of their method in accurately classifying this particular class of galaxies. Zhu et al. (2019) [19] present a study on galaxy morphology classification using deep convolutional neural networks (CNNs). They utilize a dataset of galaxy images and propose a CNN-based model for classification. Their experimental results show the capability of CNNs in accurately categorizing galaxy morphologies.

In summary, the papers discussed in this literature review encompass various approaches for galaxy morphology classification. These approaches include utilizing neural ordinary differential equations, efficientnet architectures, machine and deep learning techniques, few-shot learning, deep residual networks, and CNNs. The experimental results presented in these papers demonstrate the effectiveness of the proposed models in accurately classifying galaxy morphologies, showcasing the advancements in the field and providing valuable insights for future research in this area.

While some studies in this field have reported good results, many suffer from limitations, such as using a single dataset to show their results or relying on existing models for comparative analysis. Additionally, certain models proposed in the literature require a large number of training parameters which can lead to overfitting. To address these limitations, this study proposes a model based on the Residual Neural Network (ResNet) architecture. ResNet can achieve high accuracy while reducing computational costs, thanks to the use of skip connections. Moreover, it can generalize well to new, unseen galaxies by learning more complex features. The relatively small number of parameters in ResNet can also help to prevent overfitting, and it can be trained on various datasets, making it more representative of all types of galaxies.

### **3. METHODOLOGY**

This section provides information about the model proposed in this study, called ResNet\_Var, which uses ResNet architecture. So, before we move to ResNet\_Var, let us learn more about ResNet.

#### **3.1. Residual Neural Networks**

Residual Network (ResNet) is a Convolutional Neural Network (CNN) architecture that has revolutionized the field of computer vision by addressing the problem of vanishing gradients, which arise when a neural network becomes too deep. The backpropagation process relies on gradient descent to optimize the weights of the network. However, when there are too many layers, the repeated multiplications cause the gradient to vanish, leading to performance saturation or degradation. ResNet tackles this problem by introducing “skip connections” between layers [20]. Skip connections, also known as shortcut connections, allow information to bypass one or more layers of a neural network and be passed directly to a later layer. In ResNet,

skip connections are implemented by adding identity mappings (i.e., a simple linear function) to the output of one or more convolutional layers. In ResNet, identity mapping refers to the addition of the input of a residual block to its output. A residual block is a building block used in residual neural networks (ResNets). In a residual block, the skip connection is typically implemented as a simple identity mapping that bypasses one or more convolutional layers in the block. This means that the input to the block is added directly to the output of the block, which helps to prevent the vanishing gradient problem that can occur in very deep neural networks.

By using skip connections, residual blocks enable the network to learn features that are more abstract and complex than what could be learned without them. ResNet stacks multiple identity mappings and skips some layers by reusing the activations of the previous layer. This method speeds up the initial training phase by compressing the network into fewer layers and then expanding all layers during retraining. The term “retraining” refers to the process of fine-tuning the weights of the ResNet architecture after the initial training phase. During the initial training phase, the ResNet architecture is compressed into fewer layers by skipping some layers using skip connections. After this phase, the skipped layers are added back in, and the remaining layers are trained again on the data to improve the performance of the network. In most ResNet models, two or three layers are skipped at a time with nonlinearity and batch normalization in between [20].

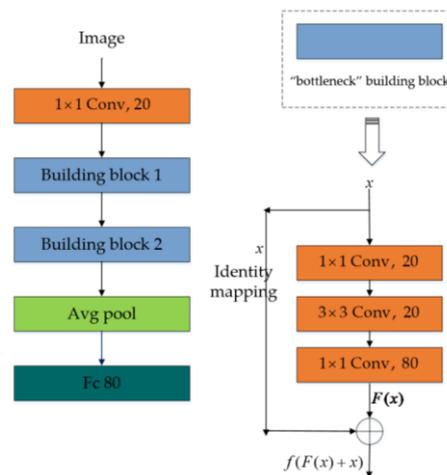


Figure 4. A residual network model architecture (Source: Fang et al. (2018) [21]).

Figure 4 illustrates the model architecture of a simple Residual Network (ResNet) and the structure of the residual block used in the ResNet model. As seen in the left part of figure 4, the image is given as input to the model, and it is passed to a convolution layer (“Conv” in the figure 4) of  $1 \times 1$  kernel with a filter size of 20. Then the image is passed through building block 1 followed by building block 2. The building block referred to in the figure 4 is also known as residual blocks. The term “bottleneck” building block in figure 4 refers to a specific type of building block that is designed to reduce the number of parameters and computations required to train a very deep neural network. Output from the building block 2 is given as an input to an average pooling layer (“Avg pool” in the figure 4), and then the output from the average pooling layer is provided as an input to the fully connected layer (“Fc” in the figure 4). Now, the right part of figure 4 shows the structure of the building block. It shows the series of convolutional layers are present inside the building block with varying kernel sizes of  $1 \times 1$  and  $3 \times 3$  as well as filter sizes of 20 and 80. As seen in the right part of figure 4,  $x$  is the input to the building block, which is the output from the previous layers in the neural network. The output from the convolutional layer, first orange rectangle with “ $1 \times 1$  Conv, 20” written inside in the left part of

figure 4, is the input for building block 1, the first blue block with “Building block 1” written inside. The value of the input to the building block,  $x$ , is kept aside to use it as identity mapping. The right part of figure 4 shows an equal input  $x$  which is not identity mapping is passed through a series of convolution layers, “ $1 \times 1$  Conv, 20”, “ $3 \times 3$  Conv, 20”, and “ $1 \times 1$  Conv, 80”. The output from the series of convolutional layers,  $F(x)$  is the residual mapping. The final output from a building block, the original mapping, is represented as  $F(x) + x$ . There are a few variants of ResNet, such as ResNet18, ResNet34, ResNet50, ResNet101, and ResNet150. The name of the variant depends on the number of convolution and pooling layers in the model. The next subsection will provide information on the model proposed in this study, called ResNet\_Var.

### 3.2. ResNet\_Var Architecture

The model architecture, ResNet\_Var, for this research is a variant of Residual Neural Network (ResNet) as shown in the Figure 5 that utilizes several techniques to improve performance and reduce overfitting. The model takes an input shape and number of classes as parameters and uses the Keras functional API to build the architecture. The shape of an image is given as  $height \times width \times colorchannels$ , and the number of classes is the number of classifications of the training dataset. To train the model for image classification, the image is passed as an input to the model, and that image passes through various layers before the image can be categorized into a class.

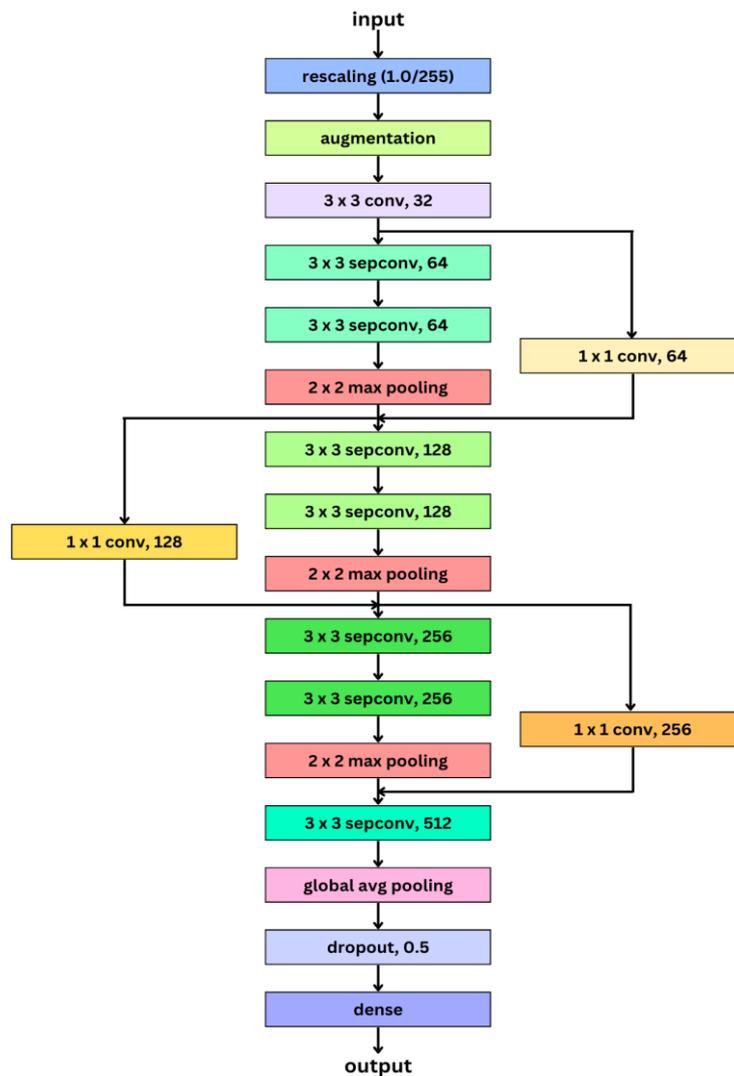


Figure 1. Architecture of the ResNet\_Var proposed in this research.

At the beginning of the training process, the input image is passed to a rescaling layer, shown in figure 5, which scales the input data by dividing it by 255. This is a common preprocessing step in image classification tasks and helps ensure that the input data is in a consistent range. Then, as illustrated in figure 5, the image passes through a data augmentation layer, which applies a set of random transformations to the input data to increase the size of the training dataset. This is important for deep ResNets, as they are particularly susceptible to overfitting when trained on small datasets. The pipeline defined in this architecture includes three different types of data augmentation: random flipping, random rotation, and random zoom. The RandomFlip layer randomly flips the input images horizontally or vertically. The RandomRotation layer randomly rotates the images by a specified angle range, in this case between 0 and 1 radians. The RandomZoom layer randomly zooms into the images by a specified range, in this case between -0.1 and -0.4. By applying these types of data augmentation, the model is trained on a wider range of data, making it more robust to different variations and improving its ability to generalize to new examples. This can lead to better performance on the test set and in real-world applications.

In figure 5, we can see that the model then uses a convolutional layer (“conv” in figure 5) which is shown as “ $3 \times 3$  conv, 32” where “ $3 \times 3$ ” is the kernel size and “32” is the filter size for the convolutional layer, which means 32 filters of size  $3 \times 3$  are used in this layer. This layer uses the ReLU activation function, which is a common choice for Neural Networks because it helps improve performance and reduce the risk of overfitting. The activation function is applied every time the image is passed through a convolutional layer. As such, it is not shown separately in figure 5. The image is passed through the residual block three times, with a different filter size in each iteration. Inside the residual block, illustrated in figure 5 (b), the model uses two Separable Convolution Layers (“sepconv” in figure 5) of kernel size  $k=3$ , with ReLU activation function and a filter size of 64, 128, and 256 in each respective iteration. In figure 5, the first separable convolutional layer is shown as “ $3 \times 3$  sepconv, 64” where “ $3 \times 3$ ” is the kernel size and “64” is the filter size for the separable convolutional layer, this structure is followed throughout the model architecture. Separable Convolution Layers are a more efficient variant of traditional convolutional layers, which is useful when working with large datasets or running the model on resource-constrained devices. The model also uses batch normalization layers to normalize the activations of the previous layer, which helps to reduce the internal covariate shift, improving performance and reducing overfitting. After the Separable Convolution layers, the model applies a max pooling layer to reduce the spatial dimensions of the input data, which reduces the number of parameters in the model and helps control overfitting. In addition to the Separable Convolution layers, the model also uses a convolutional layer of kernel size  $k=1$  in the residual connection with ReLU activation. The model uses residual connections, as shown in figure 5, to add the output of the previous block to the output of the current block, preserving information from previous layers and improving performance. The filter size for both the separable convolution layers and the convolutional layer remains the same during each iteration of the residual block, which is iterated three times with filter sizes of 64, 128, and 256.

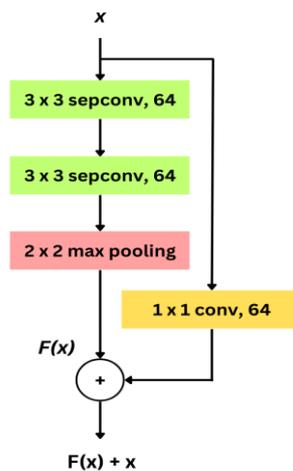


Figure 6. Structure of residual block used in ResNet\_Var.

Figure 6 shows the residual blocks used in ResNet\_Var. The residual block in ResNet\_Var uses separable convolutional layers instead of normal convolutional layers, and we can also see that the residual block of ResNet\_Var uses a max pooling layer, which can not be seen in the residual block of simple ResNet architecture shown in the right part of figure 4. The skip connection in ResNet (right part of figure 4) simply skips a series of convolutional layers, whereas the skip connection in ResNet\_Var uses a convolutional of  $1 \times 1$  kernel. The convolutional layer is added in the skip connection so that no information is ignored during the training process.

After iterating through the residual block three times with filter sizes 64, 128, and 256 in each iteration, we can see, in figure 5, that the model uses a separable convolutional layer of filter size of 512 and kernel,  $k=3$  with ReLU activation function. The model also uses global average pooling layers, which reduce the spatial dimensions of the input data to a single value. This is useful for image classification tasks, as it allows the model to focus on the global features of the input data. Finally, the model uses a dropout layer, which randomly sets some of the weights to zero during training. This helps to reduce overfitting by preventing the model from relying too heavily on any one neuron. The final output layer is a dense layer with a number of units equal to the number of classes and uses the sigmoid activation function.

The model proposed in this study, ResNet\_Var, uses separable convolution layers instead of regular convolution layers. Separable convolution layers [23] consist of a depthwise convolution layer followed by a pointwise convolution layer. This reduces the number of parameters in the model, which can make it easier and faster to train. Additionally, separable convolution layers can learn more diverse feature representations and can improve the generalization of the model. ResNet\_Var has a more diverse block structure than ResNet34 [20], a variant of the ResNet. ResNet34 uses identical blocks throughout the network, while ResNet\_Var uses blocks of varying filter sizes (64, 128, 256). This allows ResNet\_Var to capture features at multiple scales, which can improve the accuracy of the model. ResNet\_Var has a smaller number of layers than ResNet34. ResNet34 has 34 layers, while ResNet\_Var has fewer layers. This can make ResNet\_Var faster to train and can reduce the risk of overfitting.

The ResNet\_Var model proposed in this study has several advantages over the models discussed in the papers mentioned in the literature review section. First, it uses a ResNet architecture that has been shown to perform well in various computer vision tasks. Second, it uses separable convolutions to reduce the number of parameters, which helps to prevent overfitting and reduce computational costs. Third, it uses global average pooling, which reduces the number of parameters and helps to prevent overfitting. Finally, the ResNet\_Var model is relatively simple and easy to implement, making it suitable for a wide range of galaxy morphology classification tasks.

### **3.2.1. Separable Convolution Layer**

Separable convolution was introduced by Google researchers [22] to reduce the computational cost of performing convolutions in deep neural networks. A separable convolution layer in a neural network model is a type of convolution layer that performs convolution operations in a more efficient manner [23]. A regular convolution layer performs the convolution operation by applying a filter to each region of the input image, resulting in a large number of parameters that need to be learned. Separable convolutions, on the other hand, perform convolution operations in two separate stages. The first stage involves applying a depthwise convolution, which applies a separate filter to each channel of the input image. The second stage involves applying a pointwise convolution, which combines the outputs of the depthwise convolution into a single feature map (figure 7).

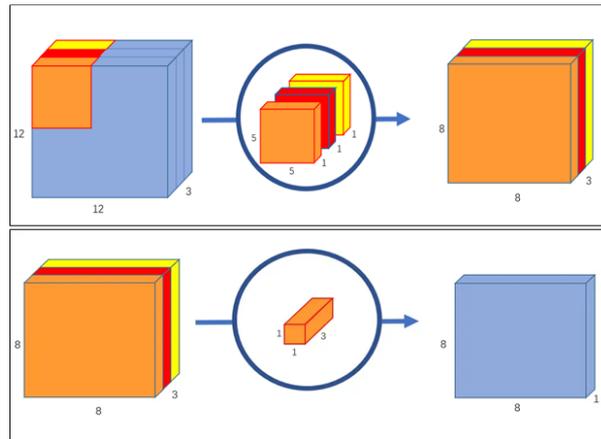


Figure 7. Depthwise convolution, uses 3 kernels to transform a  $12 \times 12 \times 3$  image to an  $8 \times 8 \times 3$  image (top), Pointwise convolution, transforms an image of 3 channels to an image of 1 channel (bottom) (Source: Wang et al. (2018) [23]).

This separation of the convolution operation into two stages results in a significant reduction in the number of parameters that need to be learned, making separable convolutions more computationally efficient than regular convolutions. Additionally, separable convolutions have been shown to produce good results in a variety of image classification tasks, making them a popular choice for many neural network models. In summary, a separable convolution layer in a neural network model performs the convolution operation in a more efficient manner by separating the operation into two stages: a depthwise convolution and a pointwise convolution. This separation results in a reduction in the number of parameters that need to be learned, making separable convolutions more computationally efficient and a popular choice for many neural network models [23].

## 4. EXPERIMENTS AND RESULTS

This research evaluated ResNet\_Var in addition to four popular Convolutional Neural Network (CNN) models - VGG16, VGG19, ResNet50, and Inception - on Dataset D1 and Dataset D2 (additional details on the datasets will be covered in section 4.1), which consist of five and seven distinct classes, respectively. The proposed model, ResNet\_Var, was trained and compared with the popular CNN models mentioned above using commonly used metrics in computer vision and machine learning, including Precision, Recall, F1-score, and Accuracy. Additionally, this study compared ResNet\_Var's performance on Dataset D1 with the models proposed by Gupta et al. (2022) [13] and Zhang et al. (2022) [16], as they also used Dataset D1. Similarly, the performance of ResNet\_Var on Dataset D2 was compared with the model proposed by Kalvankar et al. (2020) [14], as they used Dataset D2.

### 4.1. Datasets

Gathering and organizing the data is an essential part of training a neural network model. The raw data is collected from a competition launched on Kaggle in December 2013 [24]. It is a dataset, called Galaxy Zoo dataset, on Kaggle that provides information on various galaxies in the universe. The dataset contains information on the morphological properties of over 79,000 galaxies from the Sloan Digital Sky Survey (SDSS) and is sourced from the Galaxy Zoo project. The Galaxy Zoo project is a citizen science project that involves the public in classifying the shapes of galaxies from digital images.

Two subsets of the Galaxy Zoo dataset are generated after clean galaxy images are selected from the Galaxy Zoo dataset, called dataset D1 and dataset D2. The selection criteria for dataset D1 is the same as the dataset used by Gupta et al. (2022) [13] in their study, and the selection of images for dataset D2 is the same as the dataset used by Kalvankar et al. (2020) [14] in their research. Dataset D1 consists of images that have been categorized into five distinct classes based on their visual features. The categories are as follows: completely round, in-between, cigar-shaped, edge-on, and spiral, with the distribution of the images in each class as 8434, 8069, 578, 3903, and 7806 images, respectively, for each category. This dataset D2 is composed of galaxy images that have been categorized into seven distinct groups, namely completely round, in-between, cigar-shaped, lenticular, barred spiral, unbarred spiral, and irregular, with 8107, 7782, 578, 3780, 872, 3307, and 1560 images, respectively, for each category.



Figure 8. Galaxy image from each class in dataset D1, (from left to right) Cigar shaped, Completely round, Edge-on, In-between, Spiral.

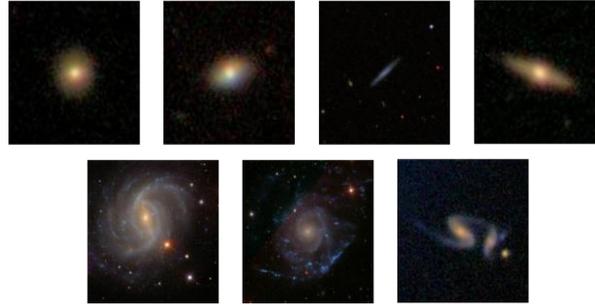


Figure 9. Galaxy image from each class in dataset D2, (from left to right) Completely round, In-between, Cigar shaped, Lenticular, Barred spiral, Unbarred spiral, Irregular.

Figure 8 shows five sample images from dataset D1 belonging to five distinct classes in dataset D1. Figure 9 shows seven sample images from dataset D2 belonging to seven distinct classes in dataset D2.

Table 1. Image distribution of Dataset D1 for training and testing.

Class No.	Galaxy Class	Training Pool	Testing Pool
0	Completely round	7,025	781
1	In-between	7,262	807
2	Cigar shaped	520	58
3	Edge-on	3,513	390
4	Spiral	7,591	843
	<b>Total</b>	<b>25,911</b>	<b>2,879</b>

After generating datasets from the GZ dataset, we need to split the data into a 9:1 ratio for training and testing the model proposed in this study, with 90% of the dataset being used for training and 10% of the dataset being used for testing. This distribution applies to both datasets, Dataset D1 and Dataset D2. Tables 1 and 2 show the distribution of images for training and testing for both datasets, D1 and D2.

Table 2. Images distribution of Dataset D2 for training and testing.

Class No.	Galaxy Class	Training Pool	Testing Pool
0	Completely round	7,297	810
1	In-between	7,004	778
2	Cigar shaped	521	57
3	Lenticular	3,402	378
4	Barred spiral	745	82
5	Unbarred Spiral	2,979	328
6	Irregular	1,404	156
	<b>Total</b>	<b>23,352</b>	<b>2,589</b>

After splitting datasets D1 and D2 for training and testing, the next step was training ResNet\_Var model and other well-known CNN models like VGG16, VGG19, ResNet50, and Inception on both datasets. The evaluation of each model's performance was done by generating a confusion matrix after testing. From the confusion matrix, we can calculate accuracy, precision, recall, and f1-score, which helps us in evaluating the performance of the model. Along with the results of popular CNN models on both the datasets, dataset D1 and dataset D2, we compared the results of ResNet\_Var, when trained on dataset D1, with the results obtained from some of the previous works such as Gupta et al. (2022) [13], Zhang et al. (2022) [16], and Zhu et al. (2019) [19] (summarized in the Literature Review section), as they all use dataset D1. And we have compared the finding of Kalvankar et al. (2020) [14] with the results of ResNet\_Var when the model is trained using Dataset D2, as they [14] used dataset D2 in their study.

#### 4.2. Less Number of Trainable Parameters

Before moving to the training and testing part, let us have a brief discussion on how the number of trainable parameters play a role in the performance of a CNN model.

Table 3. Trainable parameter of various CNN models as well as ResNet\_Var.

Models	Trainable Parameters
VGG16	14,717,253
VGG19	20,026,949
ResNet50	23,529,605
Inception	21,778,597
EfficientNetB5 [14]	28,351,029
<b>ResNet_Var</b>	<b>320,679</b>

Table 3 shows the number of trainable parameters for various popular models used for image classification in general. When the CNN models mentioned in Table 3 are imported from Keras library [26], the number of trainable parameters remains the same. It can be observed that ResNet\_Var, the model proposed in this study, has significantly fewer trainable parameters than the other models. Having fewer parameters is generally better to avoid overfitting, as the model becomes less complex and less prone to memorizing the training data. Please note that EfficientNetB5 is the model used by Kalvankar et al. (2020) [14], with over 28 million trainable parameters, which is computationally expensive, and it may lead to overfitting. The computational power required to train and run a CNN model is directly related to the number of trainable parameters in the model [25]. The more parameters a model has, the more computations it needs to perform during training and inference. So, in comparison to other CNN models, ResNet\_Var requires much less computational power, and it has less chance of overfitting.

#### 4.3. Results of ResNet\_Var Compared to Popular CNN Models using Dataset D1

To begin with, we imported several well-known Neural Network models, including VGG16, VGG19, ResNet50, and Inception, from the Keras library [26]. The Keras library is a popular open-source deep learning framework that provides access to pre-trained models and other tools for training neural networks. These models have been shown to achieve state-of-the-art performance on various computer vision tasks, such as image classification and object detection (Chollet, 2018) [27]. These models, as well as ResNet\_Var, were trained on Dataset D1, and subsequently, graphs were plotted to display the training accuracy and loss of each model. Additionally, confusion matrices were generated to evaluate the performance of the models on Dataset D1. The model proposed in this study, called ResNet\_Var, is trained on a specific dataset called D1. First, the training accuracy and loss of the model were plotted to obtain insights into the model's learning process, as shown in Figure 10.

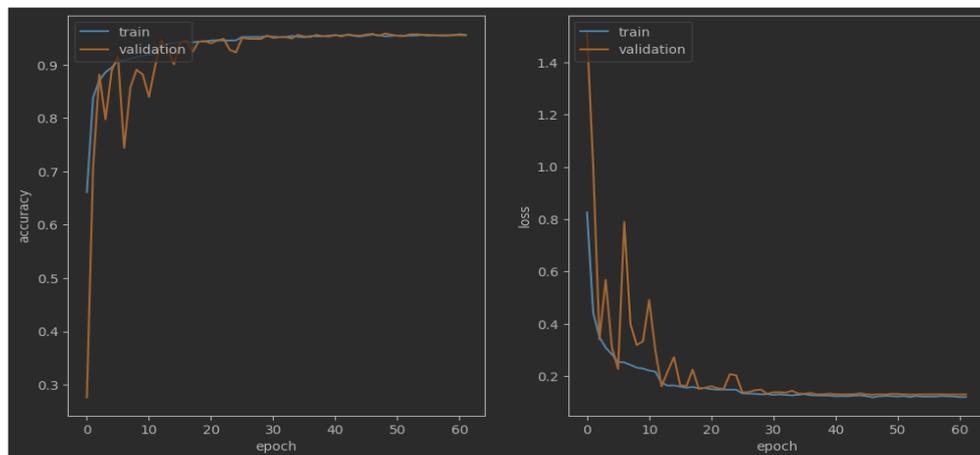


Figure 10. Training accuracy and loss, when using Dataset D1 and ResNet\_Var.

Additionally, confusion matrices were generated to further evaluate the performance of the model on D1, as depicted in Figure 11. These confusion matrices enabled the computation of key metrics such as accuracy, precision, recall, and F1 score, which are fundamental in assessing the model's performance.

The figure 12 summarizes the performance of popular CNN models on Dataset D1. The models compared include VGG16, VGG19, ResNet50, Inception, and ResNet\_Var. The evaluation metrics used are accuracy, precision, recall, and F1 score. ResNet\_Var is a proposed model that

outperforms all the other models, achieving an accuracy, precision, recall, and F1 score of 0.95 each.

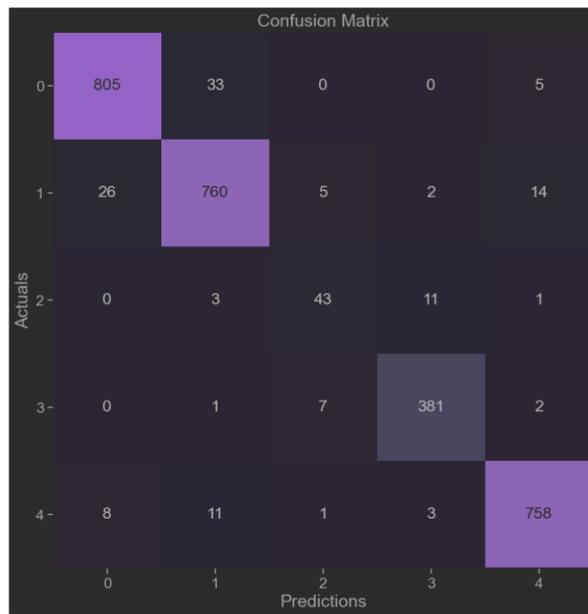


Figure 11. Confusion matrix obtained using ResNet\_Var as a result of testing data from dataset D1.

	precision	recall	f1-score
0 CompletelyRound	0.93	0.93	0.93
1 InBetween	0.91	0.92	0.91
2 CigarShaped	0.58	0.55	0.57
3 EdgeOn	0.91	0.94	0.92
4 Spiral	0.96	0.93	0.94
accuracy			0.92
macro avg	0.86	0.85	0.86
weighted avg	0.92	0.92	0.92

	precision	recall	f1-score
0 CompletelyRound	0.93	0.96	0.95
1 InBetween	0.94	0.91	0.92
2 CigarShaped	0.59	0.47	0.52
3 EdgeOn	0.90	0.96	0.93
4 Spiral	0.96	0.94	0.95
accuracy			0.93
macro avg	0.86	0.85	0.85
weighted avg	0.93	0.93	0.93

	precision	recall	f1-score
0 CompletelyRound	0.93	0.94	0.94
1 InBetween	0.92	0.91	0.92
2 CigarShaped	0.59	0.50	0.54
3 EdgeOn	0.92	0.94	0.93
4 Spiral	0.95	0.95	0.95
accuracy			0.93
macro avg	0.86	0.85	0.85
weighted avg	0.92	0.93	0.93

	precision	recall	f1-score
0 CompletelyRound	0.91	0.96	0.94
1 InBetween	0.94	0.90	0.92
2 CigarShaped	0.72	0.50	0.59
3 EdgeOn	0.92	0.96	0.94
4 Spiral	0.96	0.95	0.96
accuracy			0.93
macro avg	0.89	0.86	0.87
weighted avg	0.93	0.93	0.93

	precision	recall	f1-score
0 CompletelyRound	0.96	0.95	0.96
1 InBetween	0.94	0.94	0.94
2 CigarShaped	0.77	0.74	0.75
3 EdgeOn	0.96	0.97	0.97
4 Spiral	0.97	0.97	0.97
accuracy			0.95
macro avg	0.92	0.92	0.92
weighted avg	0.95	0.95	0.95

Figure 12. Performance of CNN models on Dataset D1 for each class. VGG16 (top left), VGG19 (top right), ResNet50 (middle left), Inception (middle right), ResNet\_Var (bottom).

Figure 12 shows the breakdown of precision, recall, and f1-score for each galaxy class of dataset D1 for VGG16, VGG19, ResNet50, Inception, and ResNet\_Var. These are the results generated after training and testing these models on dataset D1. The table seen in each sub-figure of figure 12 is known as a classification report, which is generated with the help of the Scikit-learn library [28]. In the Scikit-learn library, the weighted average is a way to compute the average of a metric across different classes in a classification task, considering the relative proportion of each class in the dataset. For example, when computing the accuracy of a model on a multi-class classification problem, we might have imbalanced classes where some classes have more samples than others. In this case, a simple average of the accuracy across all classes would not be appropriate, as it would give equal importance to each class regardless of its size. The weighted average takes into account the number of samples in each class and computes a weighted average of the metric, where the weight of each class is proportional to the number of samples in that class. Specifically, the weighted average of a metric is calculated as  $weighted\_average = \frac{\sum(weight\_i * metric\_i)}{\sum(weight\_i)}$  where metric  $i$  is the value of the metric for the  $i$ -th class, and  $weight\_i$  is the weight assigned to the  $i$ -th class, which is equal to the number of samples in that class divided by the total number of samples.

#### 4.4. Results of ResNet\_Var Compared to Popular CNN Models Using Dataset D2

Once we completed training and testing several Neural Network models on Dataset D1, we proceeded to import Convolutional Neural Network (CNN) models, namely VGG16, VGG19, ResNet50, and Inception, from the Keras library [26]. We then trained each of these models as well as ResNet\_Var using Dataset D2, another dataset we wished to evaluate the performance of the models on. Figure 13 shows plots of training accuracy and training loss for the model ResNet\_Var. Moreover, confusion matrices were generated to evaluate the performance of the models on Dataset D2, as shown in Figure 14.

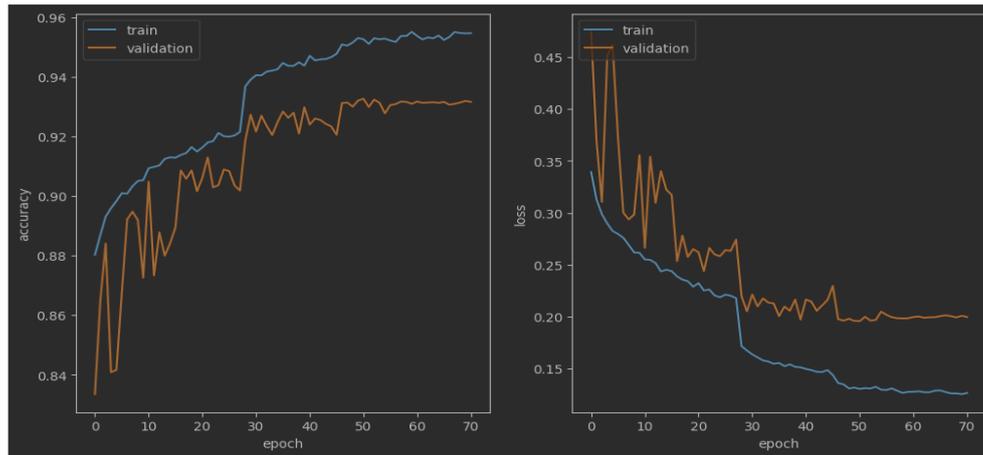


Figure 13. Training accuracy and loss obtained by ResNet\_Var, when using Dataset D2.

Figure 15 presents the performance of popular Convolutional Neural Network (CNN) models on Dataset D2 in terms of accuracy, precision, recall, and F1 score. The models included in the table are VGG16, VGG19, ResNet50, Inception, and ResNet\_Var. Figure 15 shows the breakdown of precision, recall, and f1-score for each galaxy class of dataset D2 for VGG16, VGG19, ResNet50, Inception, and ResNet\_Var. These results were generated after training and testing these models on dataset D2. The results shown in Figure 12, in section 4.3, and the results shown in Figure 15 indicate that the model proposed in this study, ResNet\_Var, outperforms popular CNN models for both datasets D1 and D2 in terms of accuracy, precision, recall, and f1-score.

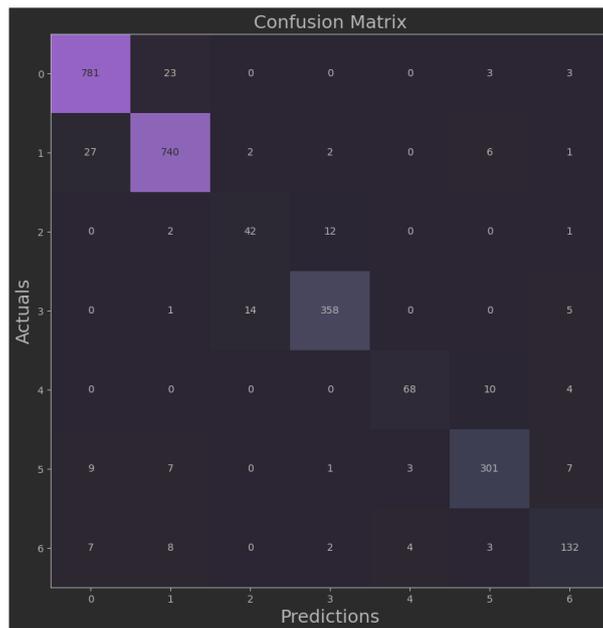


Figure 14. Confusion matrix obtained using ResNet\_Var as a result of testing data from dataset D2.

	precision	recall	f1-score
0 completely_round	0.93	0.95	0.94
1 inbetween	0.93	0.92	0.92
2 cigarshaped	0.56	0.61	0.59
3 lenticular	0.90	0.92	0.91
4 barred_spiral	0.74	0.83	0.78
5 unbarred_spiral	0.89	0.82	0.85
6 irregular	0.74	0.69	0.71
accuracy			0.89
macro avg	0.81	0.82	0.82
weighted avg	0.89	0.89	0.89

	precision	recall	f1-score
0 completely_round	0.93	0.94	0.93
1 inbetween	0.91	0.93	0.92
2 cigarshaped	0.49	0.53	0.51
3 lenticular	0.91	0.91	0.91
4 barred_spiral	0.74	0.72	0.73
5 unbarred_spiral	0.89	0.84	0.86
6 irregular	0.74	0.69	0.71
accuracy			0.89
macro avg	0.80	0.79	0.80
weighted avg	0.89	0.89	0.89

	precision	recall	f1-score
0 completely_round	0.94	0.94	0.94
1 inbetween	0.91	0.93	0.92
2 cigarshaped	0.66	0.47	0.55
3 lenticular	0.90	0.94	0.92
4 barred_spiral	0.82	0.76	0.78
5 unbarred_spiral	0.84	0.87	0.85
6 irregular	0.77	0.67	0.72
accuracy			0.89
macro avg	0.83	0.80	0.81
weighted avg	0.89	0.89	0.89

	precision	recall	f1-score
0 completely_round	0.95	0.95	0.95
1 inbetween	0.93	0.94	0.94
2 cigarshaped	0.61	0.61	0.61
3 lenticular	0.93	0.94	0.94
4 barred_spiral	0.85	0.87	0.86
5 unbarred_spiral	0.90	0.88	0.89
6 irregular	0.81	0.78	0.79
accuracy			0.92
macro avg	0.85	0.85	0.85
weighted avg	0.92	0.92	0.92

	precision	recall	f1-score
0 completely_round	0.95	0.96	0.96
1 inbetween	0.95	0.95	0.95
2 cigarshaped	0.72	0.74	0.73
3 lenticular	0.95	0.95	0.95
4 barred_spiral	0.91	0.83	0.87
5 unbarred_spiral	0.93	0.92	0.92
6 irregular	0.86	0.85	0.85
accuracy			0.94
macro avg	0.90	0.88	0.89
weighted avg	0.94	0.94	0.94

Figure 15. Performance of CNN models on Dataset D2 for each class. VGG16 (top left), VGG19 (top right), ResNet50 (middle left), Inception (middle right), ResNet\_Var (bottom).

#### 4.5. Results of ResNet\_Var Compared to Previous Works

ResNet\_Var shows promising results when the model is compared against popular CNN models, such as VGG16, VGG19, ResNet50, and Inception. After reviewing the results of popular CNN models in the previous subsection, we will obtain the results provided by a few publications which use either dataset D1 or dataset D2. Table 4 shows the results achieved by a few prior studies which focused on galaxy classification on dataset D1. Some of the publications did not mention the results in all the matrices, such as precision, recall, and f1-score. Since all the publications did include the accuracy they obtained during the classification task, we are using accuracy as a metric of comparison.

Table 4. Results obtained by prior studies which employed dataset D1 as well as results achieved by ResNet\_Var on dataset D1.

Studies	Model	Dataset	Results
Gupta et al. (2022) [13]	NODE	Dataset D1	91.62% accuracy
Zhang et al. (2022) [16]	Few-shot Learning	Dataset D1	90.90% accuracy
Zhu et al. (2019) [19]	ResNet	Dataset D1	93.12% accuracy
<b>This Study</b>	<b>ResNet_Var</b>	<b>Dataset D1</b>	<b>95.35% accuracy</b>

Table 4 shows that the model proposed in this study, ResNet\_Var, outperforms the models proposed in prior studies on dataset D1. Now, table 5 shows that ResNet\_Var produced comparable results to the model proposed in a study by Kalvankar et al. (2020) [14] on dataset D2. We need to keep in mind that the results in the last row of both Tables 4 and 5 are obtained by ResNet\_Var, which has a significantly lower number of trainable parameters when compared to other models (mentioned earlier in section 4.2).

Table 5. Results obtained by prior studies which employed dataset D2 as well as results achieved by ResNet\_Var on dataset D2.

Studies	Model	Dataset	Results
Kalvankar et al. (2020) [14]	EfficientNetB5	Dataset D2	93.70% accuracy
<b>This Study</b>	<b>ResNet_Var</b>	<b>Dataset D2</b>	<b>93.54% accuracy</b>

In conclusion, ResNet\_Var produced better results when compared against some popular CNN models as well as a few prior studies which aimed for the task of galaxy classification. It can be seen, in Table 3 (in section 4.2), that even though ResNet\_Var has significantly fewer parameters than the other models, its performance is better than all other popular CNN models. This indicates that the ResNet\_Var model has learned to represent the data efficiently despite having fewer parameters, making it a more efficient and effective model for this task. Therefore, it can be concluded that having fewer parameters is better to avoid overfitting, and the ResNet\_Var model is an effective model that has learned to represent the data efficiently despite having significantly fewer parameters than the other models.

## 5. CONCLUSIONS

The proposed model in this study outperformed existing models, including VGG16, VGG19, ResNet, Inception, and EfficientNet, in classifying galaxies from datasets D1 and D2. The model achieved a test accuracy of 95.35% for five different classes of galaxy images in dataset D1 and 93.54% for seven different classes of galaxy images in dataset D2. This study solves the problem of manually classifying galaxy images under the explosive growth of astronomical data. The

proposed model can accurately classify galaxies into different classes and significantly reduce the time and effort required for manual classification.

In future work, the proposed model can be extended to classify a larger number of galaxies from different datasets. It can also be used to classify galaxies based on other parameters such as mass, size, and color. The model can also be improved by incorporating other machine learning techniques, such as transfer learning or ensemble learning. Further research can also focus on improving the interpretability of the model to gain insights into the classification process.

## REFERENCES

- [1] Shankar, A. (2020). *Identification of orientation of galaxies in the galaxy zoo dataset using spectral clustering*.
- [2] Way, M. J., Scargle, J. D., Ali, K. M., and Srivastava, A. N. (2012). *Advances in machine learning and data mining for astronomy*. CRC Press Boca Raton.
- [3] Ivezić, Z., Connolly, A. J., VanderPlas, J. T., and Gray, A. (2014). *Statistics, data mining, and machine learning in astronomy*. In *Statistics, Data Mining, and Machine Learning in Astronomy*. Princeton University Press.
- [4] Babu, G. J. and Feigelson, E. D. (2012). *Statistical Challenges in Modern Astronomy II*. Springer Science & Business Media.
- [5] Tan, P.-N., Steinbach, M., and Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.
- [6] Conselice, C. J. (2014). *Classification of galaxies: A review*. *Annual Review of Astronomy and Astrophysics*, 52(1):291–336.
- [7] Ball, N. M., Brunner, R. J., and Myers, A. D. (2004). *Classification of galaxy images using the support vector machines*. In *Astronomical Data Analysis*, volume 3149, pages 248–257. International Society for Optics and Photonics.
- [8] Hassanien, A. M., Selim, A. M., and Tantawi, M. A. (2019). *Automatic classification of galaxy images using deep convolutional neural network*. *Journal of Ambient Intelligence and Humanized Computing*, 10(10):3961–3973.
- [9] Morales, J. C., Oostra, B., Niederste-Ostholt, M., Pellegrini, E., and Beers, T. C. (2019). *Deep learning with convolutional neural networks for the classification of galaxies*. *Monthly Notices of the Royal Astronomical Society*, 484(3):3879–3894.
- [10] Aniruddha, B., Vinod, P., Abin, J., and Nair, M. C. (2019). *Astronomical image classification using deep neural networks*. arXiv preprint arXiv:1911.07957.
- [11] Kumari, N., Gulati, S., Sharma, S., and Rani, S. (2018). *Machine learning techniques for galaxy classification*. *Journal of Physics: Conference Series*, 1054(1):012073.
- [12] Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., et al. (2008). *Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey*. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189.
- [13] Gupta, R., Srijith, P., and Desai, S. (2022). *Galaxy morphology classification using neural ordinary differential equations*. *Astronomy and Computing*, 38:100543
- [14] Kalvankar, S., Pandit, H., and Parwate, P. (2020). *Galaxy morphology classification using efficientnet architectures*. arXiv preprint arXiv:2008.13611.
- [15] Barchi, P. H., de Carvalho, R., Rosa, R. R., Sautter, R., Soares-Santos, M., Marques, B. A., Clua, E., Gonçalves, T., de S’a-Freitas, C., and Moura, T. (2020). *Machine and deep learning applied to galaxy morphology-a comparative study*. *Astronomy and Computing*, 30:100334.
- [16] Zhang, Z., Zou, Z., Li, N., and Chen, Y. (2022). *Classifying galaxy morphologies with few-shot learning*. *Research in Astronomy and Astrophysics*, 22(5):055002.
- [17] Alawi, A. E. B. and Al-Roainy, A. A. (2021). *Deep residual networks model for star-galaxy classification*. In *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, pages 1–4. IEEE.
- [18] Shetty, G. S. and Raghu, N. (2022). *The classification of satellite galaxies based on convolution neural networks machine learning algorithm*. In *2022 3rd International Conference for Emerging Technology (INCET)*, pages 1–5. IEEE.

- [19] Zhu, X.-P., Dai, J.-M., Bian, C.-J., Chen, Y., Chen, S., and Hu, C. (2019). *Galaxy morphology classification with deep convolutional neural networks*. *Astrophysics and Space Science*, 364(4):1–15
- [20] He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778.
- [21] Fang, B., Li, Y., Zhang, H., and Chan, J. C.-W. (2018). *Semi-supervised deep learning classification for hyperspectral image based on dual-strategy sample selection*. *Remote Sensing*, 10(4):574.
- [22] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. arXiv preprint arXiv:1704.04861.
- [23] Wang, C.-F. (2018). *A basic introduction to separable convolutions*.
- [24] Kaggle (2013). *Galaxy zoo: The galaxy challenge*. <https://www.kaggle.com/competitions/galaxy-zoo-the-galaxy-challenge/data>. Accessed: 2022-04-22.
- [25] Goodfellow, I., Bengio, Y., and Courville, A. (2016a). *Deep learning*. MIT Press.
- [26] Team, Keras. “*Keras Documentation: Keras Applications*.” Keras, [keras.io/api/applications/](https://keras.io/api/applications/). Accessed 6 June 2023.
- [27] Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.
- [28] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research*, 12(Oct), 2825-2830.
- [29] Willett, K. W., Lintott, C. J., Bamford, S. P., Masters, K. L., Simmons, B. D., Casteels, K. R., Edmondson, E. M., Fortson, L. F., Kaviraj, S., Keel, W. C., et al. (2013). *Galaxy zoo 2: detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey*. *Monthly Notices of the Royal Astronomical Society*, 435(4):2835–2860.
- [30] *Hubble sequence (2023) Wikipedia*. Available at: [https://en.wikipedia.org/wiki/Hubble\\_sequence](https://en.wikipedia.org/wiki/Hubble_sequence) (Accessed: 06 June 2023).
- [31] Hubble, E. P. (1926). *Extragalactic nebulae*. *The Astrophysical Journal*, 64:321.

## AUTHORS

**Jaykumar Patel** Student at University of Windsor, Ontario, Canada pursuing MSc in Computer Science.

**Dr. Dan Wu**, PhD Associate Professor at School of Computer Science, University of Windsor, Ontario, Canada