Galaxy Classification with Transfer Learning

Yilin Guo, An Hsu, & Haoyu Hu

1. Introduction

Trained on the given dataset from the Galaxy10 DeCals Dataset, which classifies galaxy images into 10 distinctive classes, we built a convolutional neural network model with the aim of classifying a new galaxy image into any one of these 10 classes. To solve the imbalance problem in the original dataset, we first applied a class weights approach. Then, we processed the data images using contrastive limited adaptive equalization and morphological transformations to improve contrast and reduce image noise. Next, we applied transfer learning using the DenseNet architecture to improve the performance of our model. Finally, we analyzed the performance of our model on the validation and test sets by plotting accuracy and loss, producing salience maps, and creating a confusion matrix.

2. Related Work

2.1 Galaxy Classification

Galaxies are massive structures composed of stars, remnants of stars, interstellar gases, dust, and dark matter, held together by gravitational forces. In 1926, Edwin Hubble introduced a classification system for galaxies based on their shapes as viewed from Earth. This system divides regular galaxies into three primary classes: elliptical, lenticular, and spiral. A fourth class is used for galaxies with irregular appearances. In 2003, Abraham et al. developed machine learning models that classify galaxies based on their physical properties, such as brightness and color. However, this approach is now less commonly used due to its potential biases. Convolutional neural networks have recently emerged as a powerful tool in image classification, including in the field of galaxy classification. By utilizing CNNs, scientists are capable of automatically extracting complex features from images and classifying large amounts of image data in a short amount of time.

2.2 Galaxy10 DECals Datasetß

Galaxy10 DECals Dataset is an enhanced version of the original Galaxy10 dataset released by the Galaxy Zoo project. The Galaxy10 dataset was created through the contribution of volunteers who classified around 270,000 images of SDSS galaxies, from which 22,000 were selected and categorized into 10 broad classes based on their shape, using the volunteers' votes. The Galaxy10 DECals dataset, on the other hand, employs images from the DESI Legacy Imaging Surveys, which offer significantly improved resolution and image quality. The dataset includes 10 broad classes of galaxies that were selected using volunteer votes with more stringent filtering criteria. The DECals dataset represents a more robust and improved version of the Galaxy10 dataset and follows the Hubble Sequence for galaxy classification.

2.3 Transfer Learning

Transfer learning has rapidly developed as a popular technique in the field of machine learning in recent years. It involves utilizing a pre-trained model to improve the performance of a new model with a smaller dataset. The fundamental idea is to transfer the knowledge learned from a large dataset on the pre-trained model to a smaller one. Essentially, pre-trained models serve as feature extractors that can be used to recognize relevant features in the new model. This approach is especially useful when the size of the smaller dataset is insufficient for effective training of a accurate model. By transferring knowledge from pretrained models, the accuracy of the new model can be improved. Additionally, training time for the new model can be reduced by directly transferring information from the pre-trained model.

2.4 DenseNet

DenseNet is a type of pre-trained CNN architecture developed in recent years. It is particularly known for its highly connected layers: every layer is connected to each other, so that every layer receives feature maps from all preceding layers. This dense connectivity not only enhance the information transfer within the model, but also reduce the risk of encountering vanishing gradient problem since all layers are connected. The complexity on the parameters is also reduced. Since each layer is connected the weights and biases associated with each neuron in the one layer are shared among all of the neurons in the previous layer; hence, less unique parameters need to be learned for each layer. We used DenseNEt-201, which has 201 layers in total.

3. Approach (and Technical Correctness)

3.1 Contrastive Limited Adaptive Equalization

Adaptive histogram equalization (AHE) is a machine learning image processing technique. It is an extension of the histogram equalization method, which aims to improve image contrast by redistributing the lightness values of pixels throughout the image. In AHE, the image is first divided into smaller regions, and the histogram equalization is applied to each region separately.

This is the main difference between classical histogram equalization and AHE; in histogram equalization, the value distribution is applied to the entire image. Therefore, the AHE algorithm is more suitable for improving the local contrast and details of the image.

In this project, we observed that most of the galaxy images have a bright center point surrounded by scattered small bright points, which are stars. Therefore, we hypothesized that emphasizing the contrast between the body of the galaxy and the universe background could possibly improve our model's performance. We then applied this processing technique as the first step in our image processing.

3.2 Morphological Transformations

Morphological transformations are simple operations that are based on the shape of an image and are usually performed on binary images. The two basic morphological operators are Erosion and Dilation, with Opening being a variant form that involves performing an erosion operation followed by a dilation operation. Research has shown that morphological openings are increasing, anti-extensive, translation invariant, and idempotent [X]. In the context of this task, the dataset is classified according to the Hubble Sequence, which is based on the morphological characteristics of galaxies. Morphological Opening is used to eliminate small, unrelated objects that are scattered around the center of the image and to focus more on the shape of the galaxy.

3.3 Data Augmentation

Data augmentation has been shown to be an effective approach for preventing overfitting and can also expand the original dataset by adding variability and flexibility to the input data. We perform data augmentation on the GPU to avoid CPU bottlenecks and GPU data starvation. For training and validation data, we use ImageDataGenerator to perform data augmentation procedures such as random rotation, random amplification, and random contrast. To avoid adding more noise to the input image, we set the filling mode as constant, which fills points outside the boundaries of the input with a constant value of 0. We do not perform data augmentation on the test data to maximize the reliability and authenticity of the test results.

3.4 Model

3.4.1 Preprocessing

We have defined a preprocessing function that will be applied to each input after the image is resized and augmented. The function first uses Contrastive Limited Adaptive Equalization to improve contrast in the images, and then applies Morphological Opening to eliminate small unrelated noise around the galaxy.

Normalization of input data has been proven to be effective in speeding up model convergence and increasing classification accuracy. To normalize image data, we rescale our images using p = p/255 (where p is the value for each pixel) to convert the pixel values into the range [0,1].

3.4.2 Class Weights

Class weights can be used during the training process when the dataset for classification is highly imbalanced, meaning that certain categories have a small probability, resulting in a relatively small number of samples in the dataset. In this task, the Cigar Shaped Smooth Galaxies category has only 334 samples, while the number of samples in other categories is around 1,900. This can lead to bias and misclassification in trained models. During the training process, we added higher class weights to the classes with fewer samples to reduce the impact of the imbalance. The loss function will pay more attention when training with data from classes with higher class weight, which can improve the robustness of classification and accelerate convergence speed. To obtain the optimal class weights for each class, we calculate them using the compute class weight function from the scikit-learn library.

3.4.3 Early Stopping

Early stopping is a form of regularization used to avoid overfitting when training a model with an iterative method. Such methods update the learner to make it better fit the training data with each iteration. In this task, we monitor the performance of the model on validation set during training and restore the best model after training. We stop the training process if the accuracy of the prediction on validation dataset is not improving for 20 epochs and restore the weights of the model that gives the highest accuracy on validation dataset.

3.4.4 Model Architecture

Since transfer learning has proven to yield good results in feature extraction, we use DenseNet201 with weights trained on the ImageNet as our pretrained model for feature extraction. After flattening the output of pre-trained model, we then add one dense layer with 1,024 neurons, which uses ReLU as the activation function. We also use drop out with 30% probability to prevent overfitting. After dropout layer, a final dense layer is added to give prediction on each class. Because our input images are quite different from the ImageNet images, we also allow the weights in the pre-trained model to be trainable.

4. Experimental Results (and Technical Correctness)

4.1 Dataset

For this project, we got the dataset from Galaxy10 DECals Dataset. Originally Galaxy 10 utilized data from Galaxy Zoo (GZ) Data Release 2, where around 270k of SDSS galaxy images were classified into 10 different classes by volunteers. Later it changed to utilize images from DESI Legacy Imaging Surveys (DECals) for better images quality and resolutions.

4.2 Model Performance

For our experiments, we trained three models which have different pre-trained feature extraction model but the same classification model for maximal 50 epochs with the batch size of 32, using Adam as optimizer and standard categorical cross-entropy as loss function.

For our first model, we use VGG19 with non-trainable weights from ImageNet as backbone model. This model has 12,850,698 trainable parameters and converges after 14 epochs, which takes about 65 minutes. It performs poorly on validation dataset, with an accuracy score of 29.1.

For our second model, we use DenseNet201 with non-trainable weights from ImageNet as backbone model. This model converges quickly after 6 epochs within 30 minutes. During training process, we do not use weighted classes to balance the training dataset, thus the model performs poorly on minority classes such as class 0 and class 4. In the classification report on validation dataset, the f1-scores for class 0 and class 4 are both 0, and the total accuracy is 47.51%, which is much better than the previous model. To analyze the model, we generate the saliency map and Grad-CAM for images from class with highest f1-score, we discover that the model focuses too much on the surrounding instead of the center of the image. This may cause by the fact that our input images of galaxies are quite different from the ImageNet images, so the pretrained model can hardly extract proper features for classification.

For our third model, we still use DenseNet201 with pre-trained weights from ImageNet as the backbone model, but we set its weights to trainable. This model converges after 43 epochs. During training process, we use weighted classes to deal with imbalanced training dataset, thus the model performance improves significantly on minority classes such as class 0 and class 4. In the classification report on validation dataset, the f1-score for class 0 reaches 51.3% and that of class 4 reaches 62.9%, and the total accuracy is 73.13%, which is much better than the previous models. And on test dataset, the model achieved an accuracy of 78.47%, almost 30 percent higher than the second model.

4.3 Case Study

As shown in Figure N, it is evident that the model tends to confuse classes 0, 2, and 6. Due to the large number of layers and parameters, the model is challenging to interpret. We cannot simply plot the distribution of weights or different kernels to explain which features our model relies on. However, Grad-CAM and Saliency Map can reveal to some extent which features or areas most influence the prediction result. To identify where our model performs poorly, we investigated some error cases. By generating the Saliency Map, we found that most of the time, it focuses on the correct part for classifying the image instead of the noises around it. In some error cases, as shown in Figure M, the model focuses on the noisy surroundings, which leads to incorrect predictions.

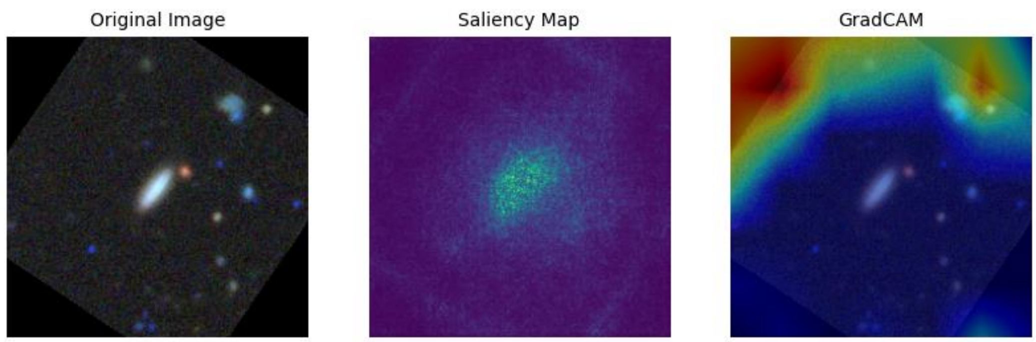

Figure 1. Figure M.

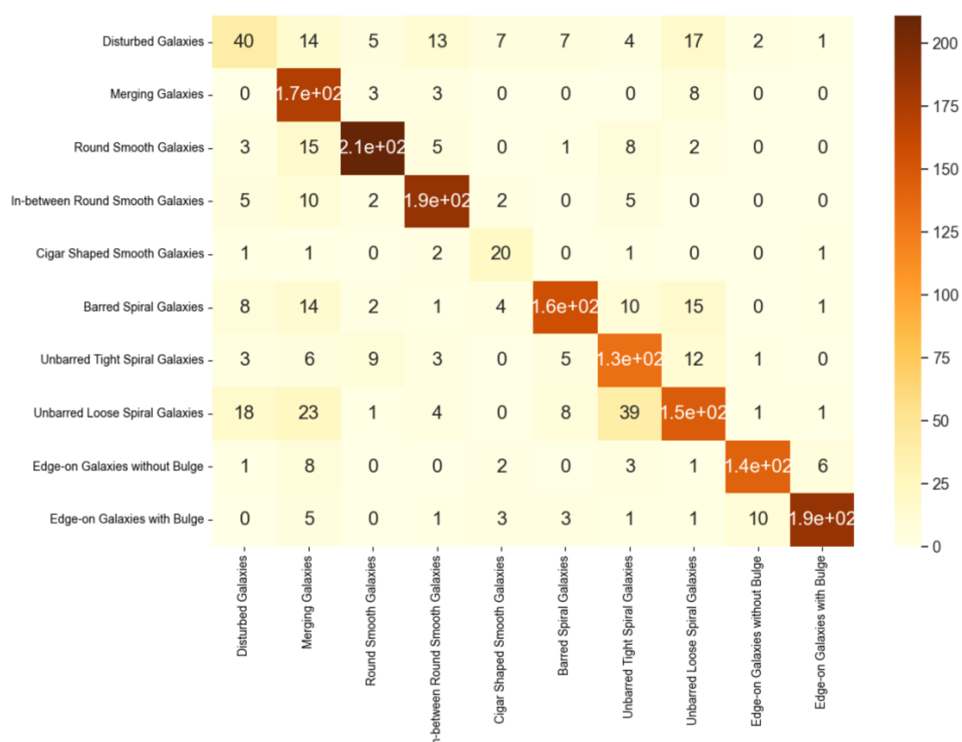

Figure 2. Figure N.

5. Conclusion

To conclude our project, we presented a CNN model for the galaxy image classification problem. The main idea behind this CNN model design is based on transfer learning. We used DenseNet201 as the pre-trained model for feature extraction and then built our own model based on that. To analyze our model's performance, we used saliency maps and Grad-CAM to explain which part of the features influences the model's prediction the most. The accuracy of our model is 78.47.

Due to time constraints, we were unable to try more variations of the model and improve its accuracy. For future improvement, different pre-trained models should be tested to see if they work better on this dataset. Overall, our model successfully demonstrates how a CNN model can accomplish a galaxy classification task via transfer learning.

References

Galaxy10 decals dataset (2023) Galaxy10 DECals Dataset - astroNN 1.2.dev0 documentation. (Accessed: 11 May 2023).

Team, G.L. (2022) What is histogram equalization and how it works?, Great Learning Blog: Free Resources what Matters to shape your Career! Available at: https://www.mygreatlearning.com/blog/histogramequalization-explained/ (Accessed: 11 May 2023).

Tsang, S.-H. (2019) Review: DenseNet - dense convolutional network (image classification), Medium. Available at: https://towardsdatascience.com/review-densenet-imageclassification-b6631a8ef803 (Accessed: 11 May 2023).