Galaxy Morphology Classification Based On An Improved Deep Convolutional Neural Network

Jiawei Chen*
Cranbrook Schools, Bloomfield Hills, Michigan, USA

*Corresponding author e-mail: hzhe_wjdi@163.com

Abstract. With each passing day telescopes around and above the Earth capture more and more images of distant galaxies. As better and bigger telescopes continue to collect these images, the datasets begin to explode in size. In order to better understand how the different shapes (or morphologies) of galaxies relate to the physics that create them, such images need to be sorted and classified. Combining with Kaggle public datasets called Galaxy Zoo, this paper devised a deep convolutional neural network for galaxy morphology image classification. The network contains eight convolutional layers, five max_pooling layers, one flatten layer, a fully connected layer contains 150 neurons, as well as the final layer which outputs galaxy category of probability distribution of each image. The experiment results show that our neural network on the validation set of cosine similarity reached to -0.8652 (the closer to -1 means that the closer the predicted output and the expected output), implying that our model for galaxy morphology classification is very effective.

1. Introduction
Understanding how and why we are here is one of the fundamental questions for the human race. Part of the answer to this question lies in the origins of galaxies, such as our own Milky Way. Yet questions remain about how the Milky Way (or any of the other ~100 billion galaxies in our Universe) was formed and has evolved. Galaxies come in all shapes, sizes and colors: from beautiful spirals to huge ellipticals. Understanding the distribution, location and types of galaxies as a function of shape, size, and color are critical pieces for solving this puzzle.

The classification method of galaxy morphology based on traditional machine learning requires complex feature engineering and exquisite design and extraction of features by domain knowledge. The quality of feature extraction directly determines the classification performance of the final classifier, which is highly subjective. Deep learning can automatically find complex and effective high-order features, avoiding the tedious process of manually extracting features and the classification problems caused by subjectivity [5,6,7]. Based on the comparison and analysis of the classical Convolutional Neural Network (CNN) including AlexNet, VGG, Inception and ResNet[9], this paper proposes a Galaxy Morphology Classification method based on an improved Convolutional Neural Network.
2. Materials

2.1. Galaxy Datasets

Galaxies in this set [1] have already been classified once through the help of hundreds of thousands of volunteers, who collectively classified the shapes of these images by eye in a successful citizen science crowdsourcing project. In our project, we use two important files: 1. images_training: JPG images of 61758 galaxies, files are named according to their GalaxyID; 2. solutions_training: Probability distributions for the classifications for each of the training images. Some galaxy images in the training set are as follows:

Figure 1. Galaxy images in the training set, galaxy ids from left to right are 10008, 10009, 10010, respectively.

The first column in each solution is labeled GalaxyID; this is a randomly-generated ID that only allows you to match the probability distributions with the images. The next 37 columns are all floating point numbers between 0 and 1 inclusive. These represent the morphology (or shape) of the galaxy in 37 different categories as identified by crowdsourced volunteer classifications as part of the Galaxy Zoo 2 project. These morphologies are related to probabilities for each category; a high number (close to 1) indicates that many users identified this morphology category for the galaxy with a high level of confidence. Low numbers for a category (close to 0) indicate the feature is likely not present.

Figure 2. The first column in each solution is labeled GalaxyID. The next 37 columns are all floating point numbers between 0 and 1 inclusive, represents Probability distributions for the classifications for each of the training images.

2.2. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN), artificial neurons can cover a surrounding unit in a corresponding part, and have excellent performance for large image processing. This network includes...
a convolution layer and a pooling layer. This network consists of an input layer, a convolution layer, an activation function, a pooling layer, and a fully connected layer. Convolution refers to the extraction of features on the original input, which extraction is simply a small fish in the original input to extract features.

![Fully connected neural network](image)

**Figure 3.** Fully connected neural network

### 2.3. Galaxy Images Metrics

In our project, the final output is the probability distribution of each galaxy image in 37 galaxies, which can be recorded as the vector y (output), and the corresponding image is also a vector, also in the 37 galaxies and the probability distribution can be written as y(label). To compare the similarity between y(output) and y(label), we use the cosine similarity in the metrics module that comes with Keras. The principle of cosine_proximity is as follows:

Let vector \( A = (A_1, A_2, \ldots, A_n) \) and \( B = (B_1, B_2, \ldots, B_n) \), then

\[
\cos \theta = \frac{\sum^n_{i=1} (A_i \times B_i)}{\sqrt{\sum^n_{i=1} A_i^2} \times \sqrt{\sum^n_{i=1} B_i^2}}
\]

(1)

The cosine_proximity \( \varepsilon \) can be represented as:

\[
\varepsilon = -\frac{1}{m} \sum^m_{i=1} \cos \theta
\]

(2)

The cosine_proximity value ranges between \([-1, 1]\), by studying the source code of cosine_proximity in the deep learning framework Keras. We found that the closer the value is to -1, the more consistent the two vectors are. When the value is equal to 0, the two vectors are orthogonal. When the value is equal to 1, the two vectors are The reverse is exactly the opposite.

### 3. Experiment Design

#### 3.1. Model Architecture

We designed a deep convolutional neural network with a total of 16 layers, including 8 convolutional layers, 5 layers of max_pooling layer, 1 flatten layer and the last two fully connected output layers. The last layer of the output layer uses the Sigmoid activation function like above the Fig4. The model structure is as follows:
Figure 4. Our Designed Deep Convolutional Neural Network Architecture

The color of our galaxy image is RGB, and its dimension is (424,424,3). We use numpy array to convert the image into an array to input into our deep convolutional neural network.

4. Data Analysis
After 15 epoch training is completed, we get the following results:

Figure 5. model loss on Train&Validation datasets

From Figure 5, we can see that the train loss is significantly lower in the first two epoch, and the subsequent epoch is more smoothly reduced, mainly because the neural network is exposed to more and more data, and its image of the galaxy Feature learning is more and more accurate, and naturally the actual output is getting closer and closer to the expected output. In the subsequent epoch, the neural network is fine-tuned, and the expected output is continuously approached, and finally it converges. The validation loss is consistent with the train loss. It is well understood that as the weight of the neural network is adjusted, the performance on the validation set is closer to the expected output.

At the same time, we also get cosine proximity, the indicator metric output vector (the probability distribution of the input galaxy image in 37 categories of galaxy) and the expectation vector, that is, the probability that the galaxy image we mark belongs to each of the 37 categories, the result is as follows:
Comparing Figure 5, we find that Figure 6, whether in the overall downward trend until the final convergence, or in the local decline, is consistent. The cosine proximity decreases toward -1, indicating that the label vector is more consistent with the output vector, and the final two vectors have a similarity of 86.52%.

5. Conclusion and Discussion
Based on the improved deep convolutional neural network, this study realizes the automatic classification of galaxie\textsuperscript{s} morphology, and the effect is good. It solves the problem that manual classification is difficult to implement under the explosive growth of astronomical data, and it is a successful application of deep learning method in the field of galaxy image.

In the future, we can try a better deep learning model to further improve the effectiveness of classification, and make product-level applications, such as mobile terminal App, to meet more astronomy enthusiasts to identify the shape of the galaxy or greatly reduce the workload of astronomical researchers on the classification of galaxies.

References
[1] Fang Guanwen; Ma Zhongyang; Kong Xu.; Study on Morphological Classification of High Redshift Mass galaxies in COSMOS Field[J]; Journal of Astronomical Sciences; 2015-05
[2] Fang Guanwen; Ma Zhongyang; Chen Yang; Kong Xu.; Evolution of stellar star formation in COSMOS field[J]; Journal of Astronomy; 2015-01
[3] Xu Fangfang; Viewing the mysterious universe The mystery of solving the galaxy——Record Shi Yong, a professor at the College of Astronomy and Space Science, Nanjing University[J]; Science Chinese; 2017-13
[4] Wang Min; Kong Xu; Research Progress in the Classification of Galaxies[J]; Progress in Astronomy; 2007-03
[5] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprintarXiv:1409.1556, 2014.
[6] Hinton, G.E., “Deep Belief Nets,” 2017 NIPS Tutorials Notes, Neural Information Processing Systems Conference, Vancouver, BC, December 2007.
[7] LeCun, Y., Jackel, L.D., Boser, B., Denker, J.S., Graf, H.P., Guyon, I., Henderson, D., Howard, R.E., and Hubbard, W., “Handwritten Digit Recognition: Application of Neural Network Chips and Automatic Learning,” IEEE Communication Magazine, 27(11), 41-46, 1989.
[8] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.
[9] P. Ekman, He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C].
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[10] Kohonen T. An introduction to neural computing[J]. Neural networks, 1988, 1(1): 3-16.