GalaxSee: Classification of the Galaxy Zoo Data Set

Vaishnav V. Rao, Harshda Saxena, Nabeel Ahmed, Friederika Biffar

***Abstract*—Galaxies can be grouped into classes based on features present in their telescope images. The Galaxy Zoo is a dataset in which galaxies have been classified by human volunteers. We develop a machine learning model that reproduces the probability distributions derived from these human classifications. We achieve the same using Convolutional Neural Networks (CNNs).**

***Index Terms*—Machine Learning, Image Classification, Convolutional Neural Networks, Galaxy Zoo**

I. Introduction

Galaxies are systems of stars, dust and gas held together by gravitational attraction and have always been of fundamental interest in astronomy and astrophysics. No two galaxies are the same, yet they can be grouped into classes based on similarities in their features such as shape, size, spirality etc.

With the advancement of space technologies, telescope images of new galaxies are readily obtained on a regular basis, and the number of new galaxies discovered has exploded to billions. It is important to quickly classify these new galaxies and obtain labels with high accuracy.

This task has historically been carried out by humans. However, it can easily be carried out by a well-trained neural network provided that a huge data set of groupings is available. The Galaxy Zoo Dataset is one such dataset that has been prepared by Kaggle, Zooniverse and Winton Capital through citizen science crowdsourcing methods.

This dataset contains thousands of images of galaxies and the probability of each galaxy belonging to a particular class has been evaluated by human sight. We train a Convolutional Neural Network on this dataset and present our results below.

II. Image characteristics

The Galaxy Zoo dataset is based on 64,000 JPG images. Each image has dimensions 424 x 424 pixels and has a size of 16 KB. These images have been obtained through telescopes in Hawaii and Chile as part of the Sloan Digital Sky Survey (SDSS) and have been compiled by the agencies mentioned above [1]. Each galaxy has a unique Galaxy ID which acts as an identifier. Some sample images are provided in Figure 1.

Fig. 1. Images from Galaxy Zoo

III. Description of the Data Set

The Galaxy Zoo data set was prepared by humans by sight and the probabilities were calculated accordingly by statistical averaging.

For each galaxy, eleven questions were asked. These are listed below along with possible responses:

1) Is the object a smooth galaxy, a galaxy with features/disk or a star? (3 responses)
2) Is it edge-on? (yes/no)
3) Is there a bar? (yes/no)
4) Is there a spiral pattern? (yes/no)
5) How prominent is the central bulge? (no bulge, just noticeable, obvious, dominant)
6) Is there anything "odd" about the galaxy? (yes/no)
7) How round is the smooth galaxy? (completely round, in between, cigar shaped)
8) What is the odd feature? (ring, lens/arc, disturbed, irregular, merger, dust lane, other)
9) What shape is the bulge in the edge-on galaxy? (rounded, boxy, no bulge)
10) How tightly wound are the spiral arms? (tight, medium, loose)
11) How many spiral arms are there? (1,2,3,4,>4,can't tell)

Thus there exist 37 possible probability values that can be associated with each galaxy. The probabilities of the responses to each of the eleven question sum up to one. The data set, therefore, consists of target vectors which are the 37 tuples corresponding to each Galaxy ID.

This Galaxy Zoo decision tree is highlighted in Figure 2.

IV. Description of the Tasks

Our workflow is divided into four tasks

1) Image visualisation and preprocessing
2) Feature extraction followed by training usual models such as Random Forest and a basic Neural Network.
3) Directly loading the images into a model and obtaining the desired 37 target probabilities in two ways: one using a basic Keras-based CNN and one using the Deep Learning network ResNet50.
4) Altering the target vectors for training by reducing the 37 target probabilities to 5 class labels, and using a basic CNN to carry out classification on the same.

V. Image visualisation and Preprocessing

The 'PIL' and 'sci-kit-image' library in python were used to pre-processes and displays the images according to their corresponding Galaxy IDs. The pre-processing steps are as follows:

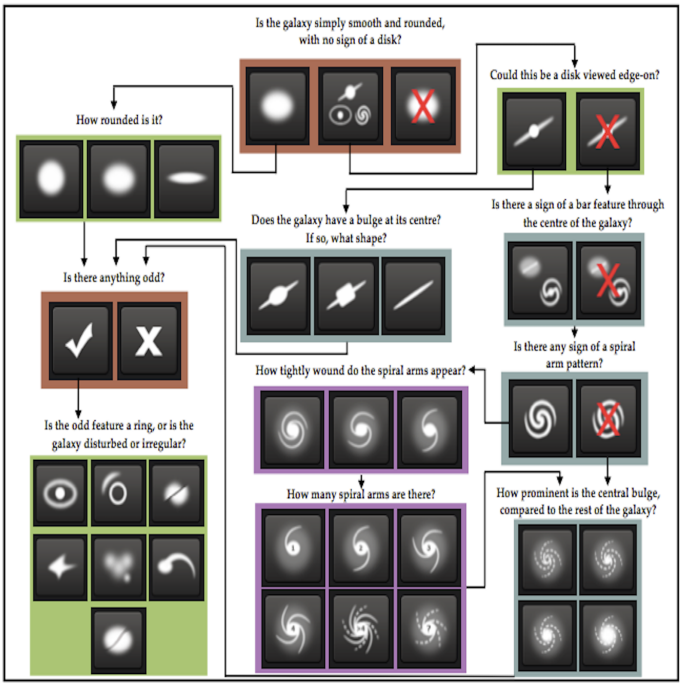

Fig. 2. The Galaxy Zoo decision tree (taken from [2])

- Crop the 424x424 images at the center to obtain 256x256 images
- Resize the 256x256 images to 64x64 images

VI. Feature Extraction using Resnet 18

First, we extracted 512 features from each image using Resnet18 as a fixed feature extractor as described in the Pytorch Transfer Learning Tutorial. These features were saved in a CSV. We used these features to train a random forest (with the best hyperparameters of the number of nodes and depth as found by GridSearchCV), which gave us high training accuracy of 85% but low validation accuracy of 33% as expected due to overfitting. After this, we trained a basic neural network with a linear fitting, which gave low MSE loss, implying the need to pursue neural networks and CNNs in more detail for better accuracy.

VII. CNNs and Resnet50

A. Using a basic Keras-based CNN

The Keras API written in Python provides progressive disclosure of complexity in which models can be stacked on top of each other. After appropriately cropping our images, we use Keras to import a sequential model, to which we add our convolution layers. Our initial model includes a 2D convolution layer of 512 channels, and (3,3) filter size with RELU activation. Then, we add a max pooling layer to decrease the size by half and finally flatten the array into 1D using a global max pooling. Finally, we add a dense fully connected layer for 37 outputs with sigmoid activation and use cross entropy for the loss. We get an accuracy of 60% training and validation. In the second model, we add more samples, three convolutional layers, and max-pooling, and additionally use two dropout layers for increased regularization. Batch gradient descent was performed using batches of 128 over 20 epochs with the loss function as binary cross entropy and the optimizer as adamax. Even after these additions, we get a very similar accuracy of 60% to the model above which thus indicates that increasing the complexity of the model may not be the best solution.

B. Using ResNet50

We then tried another model to obtain the 37 probabilities using regression. The preprocessed images were directly passed through a modified Resnet50 model with pre-trained weights set to those of 'IMAGENET' and the last two layers modified such that the last dense layer has 37 outputs with sigmoid activation. Batch gradient descent was performed using batches of 128 over 20 epochs with the loss function as binary cross entropy and the optimizer as adamax. There was no significant improvement in the accuracy as it remained roughly around 60%.

VIII. Classification using labeled images

In the previous sections, training the different neural network models to output the 37 probabilities required for each galaxy image could have been a task that required much more complex neural networks, the computation time to which we did not have access. Thus, we now attempt to simplify the problem and train models to classify galaxies that have a specific set of labels attached to them.

A. Assigning labels to each image

Based on the probability scores attached to each question and its different possible answers, we use the following set of conditions to classify the galaxies into completely rounded, rounded in-between, cigar-shaped, edge-on, and spiral.

- completely rounded: smooth (number of votes > 0.469), completely_round (> 0.469)
- rounded in-between: smooth (> 0.469), rounded_in_between (> 0.5)
- cigar-shaped: smooth (> 0.469), cigar_shaped (> 0.5)
- edge-on: features_or_disk (> 0.43), edgeon_yes (> 0.602)
- spiral: features_or_disk (> 0.43), edgeon_no (> 0.715), spiral (> 0.619)

Here smooth, completely_round, rounded_in_between, cigar_shaped, features_or_disk, edgeon_yes, edgeon_no, and spiral correspond to certain columns in the target matrix. There were $\sim$ 29000 which fell into one out of these five classes. With these five classes assigned, one-hot bit encoding of the target vector with the five classes was performed and the data was split in an 8:2 ratio for training and validation

B. Classifying using a basic CNN

Initially, we tried to train the last two layers of Resnet50 for this classification task. However, due to the excessive computation time, we abandoned this approach and set up a basic

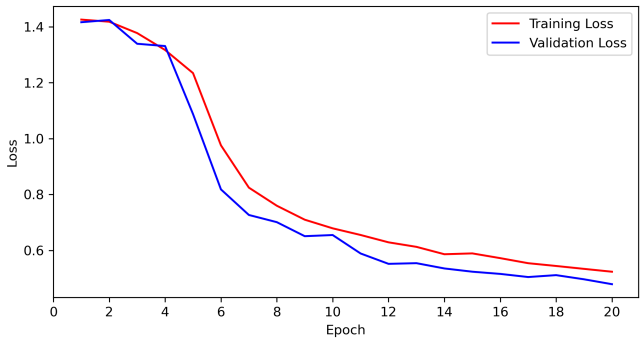

Fig. 3. Training and validation loss after each epoch of training

CNN with three convolution layers, three max-pooling layers, and a few dropout and dense layers such that the final layer had five outputs (for the five classes) with softmax activation. The loss function used was categorical cross-entropy, and the optimizer used was 'Adaptive Moment Estimation (ADAM)'. This particular optimizer was chosen as it uses adaptive learning rates for each parameter and adaptive moments to estimate first and second moments of the gradients, giving it a robustness to noise and improving convergence, compared to the traditional Stochastic Gradient Descent (SGD). We further used a batch size of 128 and went through 20 epochs of training. The results obtained were:

- Loss: 0.5236
- Accuracy: 0.7992
- Validation loss: 0.4790
- Validation Accuracy: 0.8193

The training and validation loss after each training epoch has been plotted in Figure 3. A good learning rate with a flattening of the loss can be seen in the same. So far, this is our most promising model as it gives a validation accuracy of 81%.

IX. Results

1) **For classification based on probabilities:** Deploying a random forest gives accuracy of 33% which is attributed to overfitting. Using a basic Keras-based Convolutional Neural Network improves performance accuracy to 60%. ResNet50 also provides an accuracy to 60%.
2) **For classification based on image labels:** Using a basic CNN provides an accuracy of 81%

X. Discussions

- It is observed that simply using one layer of extracted features followed by one model provides accuracy that is worse than chance, as seen in our random forest model. This indicates the need for using weights from multiple layers and hence the need for neural networks.
- For the same data set, if target vectors are tuples of floating values, then the accuracy is lower as compared to when the target vectors are simple labels. In other words, classification of labels has been carried out more accurately as compared to classification based on probabilities. This could be attributed to the fact that in the latter the machine has to produce a target vector that consists of a set of highly variable values, whereas when we modified our problem to the former, our target output was a tuple of binary digits which makes it mathematically simpler. However, the number of classes in regression was 37 whereas the number of classes in classification was 5, which does not allow us to ascertain conclusively.

Acknowledgments

We owe our sincerest thanks to Prof. Amit Sethi for his continued guidance and ideas about how to improve our project. We also thank Mukta Wagle for her inputs on ResNet50.

References

[1] Chris J. Lintott, Kevin Schawinski, Anže Slosar, Kate Land, Steven Bamford, Daniel Thomas, M. Jordan Raddick, Robert C. Nichol, Alex Szalay, Dan Andreescu, Phil Murray, and Jan Vandenberg. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. , 389(3):1179–1189, September 2008.

[2] Kyle W. Willett, Chris J. Lintott, Steven P. Bamford, Karen L. Masters, Brooke D. Simmons, Kevin R. V. Casteels, Edward M. Edmondson, Lucy F. Fortson, Sugata Kaviraj, William C. Keel, Thomas Melvin, Robert C. Nichol, M. Jordan Raddick, Kevin Schawinski, Robert J. Simpson, Ramin A. Skibba, Arfon M. Smith, and Daniel Thomas. 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, sep 2013.