Galaxification Project Report

Naga Sumanth Vankadari (Email: nagasumanthvankadari@my.unt.edu)

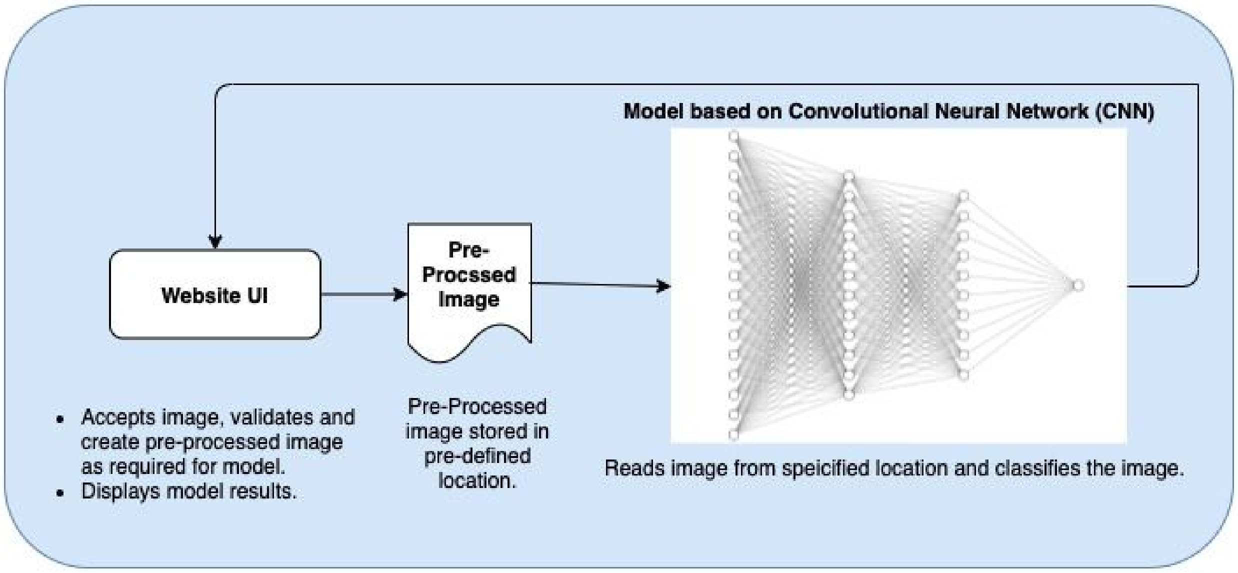
Abstract

The classification of galaxies has traditionally been accomplished manually, either by scientists or by crowd-sourcing means. However, with current technology such as the Hubble telescope, we have more unclassified images of galaxies than ever before and are in need of a better method to solve this issue.

We intend to train a machine learning model, specifically a convolutional neural network, that is capable of classifying new images of galaxies. We will use the 'Galaxy Zoo' dataset found on astroNN.readthedocs.io, which consists of 21,785 labeled images that we can train and test our model with. Our initial plans included testing out other potential options and graphically comparing the results, however we ultimately decided to use Convolutional Neural Networks (CNN) for our model and goals.

Once our model selection and training phase is complete, we will implement a simple website in which new images can be passed through and receive a classification. In order to make the website more user friendly, we will add in an image pre-processor. This allows the user to input images of various sizes and types, which will be converted to fit our model.

Overview


Model based on Convolutional Neural Network (CNN)
Website UI
Pre-Procssed Image
Accepts image, validates and create pre-processed image as required for model.
Displays model results.
Pre-Processed image stored in pre-defined location.
Reads image from speicified location and classifies the image.

Data Specification

The data set comes from astroNN's Galaxy10 documentation designed for classifying images of galaxies. There are 21,785 .jpg images in the data set. Features we considered in the dataset include color, shape, and size. The classes refer to the following labels:

- Class 0: Disk, Face-on, No Spiral
- Class 1: Smooth, Completely round
- Class 2: Smooth, in-between round
- Class 3: Smooth, Cigar shaped
- Class 4: Disk, Edge-on, Rounded Bulge
- Class 5: Disk, Edge-on, Boxy Bulge
- Class 6 : Disk, Edge-on, No Bulge
- Class 7: Disk, Face-on, Tight Spiral
- Class 8: Disk, Face-on, Medium Spiral
- Class 9: Disk, Face-on, Loose Spiral

Due to size and time constraints, we split our model into randomly selected batches of 25 epochs.

In order to deal with the above issue, we set up 11 networks, split the classes apart from the training data, pool together the predicted values, and output a CSV file full of results. We were apprehensive about the amount of time this method would take, especially before the final presentation, so as an alternative we also created a decision tree model that goes through the outputs and decides the shape (spiral, elliptical, irregular). We wanted to let the CNN model attempt to correlate features in the images to one value of one class instead of to all values of all classes. Ultimately we decided on using the 11 network output set up.

One of the specific supervised learning problems we ran into so far is training accuracy. Our model appears too accurate, consistently achieving levels between 98% to 100% which makes us suspicious. We fortunately managed to identify and test new solutions onto the issue, such as importing the data in grayscale and reducing resolution sizing to 207 x 207. While our accuracy has wavered here and there a little, our outputs appear to carry the desired results.

Results

Kaggle Galaxy Dataset:

Our original plan was to use the dataset from the Kaggle competition, however we decided not to continue with this dataset for a few reasons. The main reason is that the labels are in a decision tree format for eleven separate qualifying questions. Although there is a decision tree example on Kaggle, it does not give an end classification. This means that we are not given labels such as “Spiral galaxy” or “Elliptical galaxy”, but rather we are given floating point numbers which stand for a probability of the likelihood for each question. Our end goal for the project is to allow the user to input an image of a galaxy and to get a classification as a result, not a list of floating point numbers.

We attempted to solve this by creating our own decision tree and giving sparse labels to each image. However this is a rather difficult task given that we must decide what the label should be based on the numbers given. While we felt confident on certain decisions, others left us in doubt. If the labels we determine are not correct, then the model will be incorrectly trained and our overall performance will suffer.

Another issue lay in the dataset itself. It consisted of 61,578 jpg images of size 424x424, all of which are in color. This is simply too big to run on our computers, but taking a sample from them may not provide us with an appropriate training set. To combat this issue, we cropped the outside of the images, as each image has the galaxy centered with a lot of space around it. Next we downsampled the image to the size of 64x64 in order to more efficiently store the images. While this may decrease the accuracy slightly, we felt it was worth doing. We also experimented with converting the images to grayscale, though we didn’t see much of a performance difference here.

In the end, our model had a performance of around 63% training accuracy and only 52% validation accuracy, however we decided that due to the lack of confidence in our ability to accurately label the data based on the given decision tree, it is better to not use this dataset for our model.

Galaxy10 Dataset:

Instead we decided to go with the Galaxy 10 Dataset. This is actually a subset of the dataset from Kaggle, however there are already true labels given. This allowed us to work with the images from the dataset without fear of mislabeling them and disrupting our training.

After training our model on this dataset we saw a significant increase in accuracy, both in training and in validation as referenced in figure 1. We can see that over 25 epochs we get a training accuracy of 79% and a validation accuracy of 78%. We are still tuning the hyperparameters to test for better performance, but for now this is acceptable for our purposes. We also observed a general decrease in loss over every epoch, as shown in figure 2.

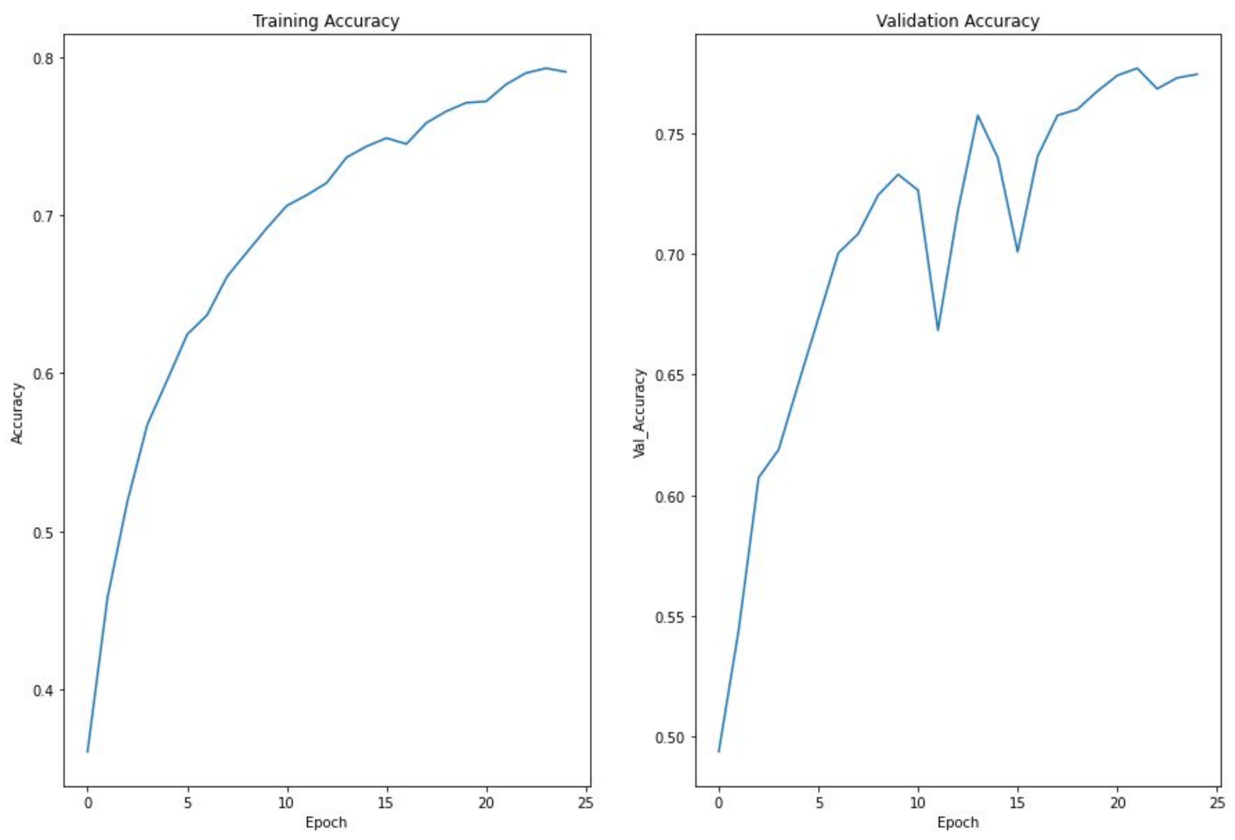


Figure 1: Training and Validation Accuracy Over 25 Epochs
On the left we observe an upward trend in training accuracy. On the right we observe more dips in the validation accuracy, but with an upward trend regardless.

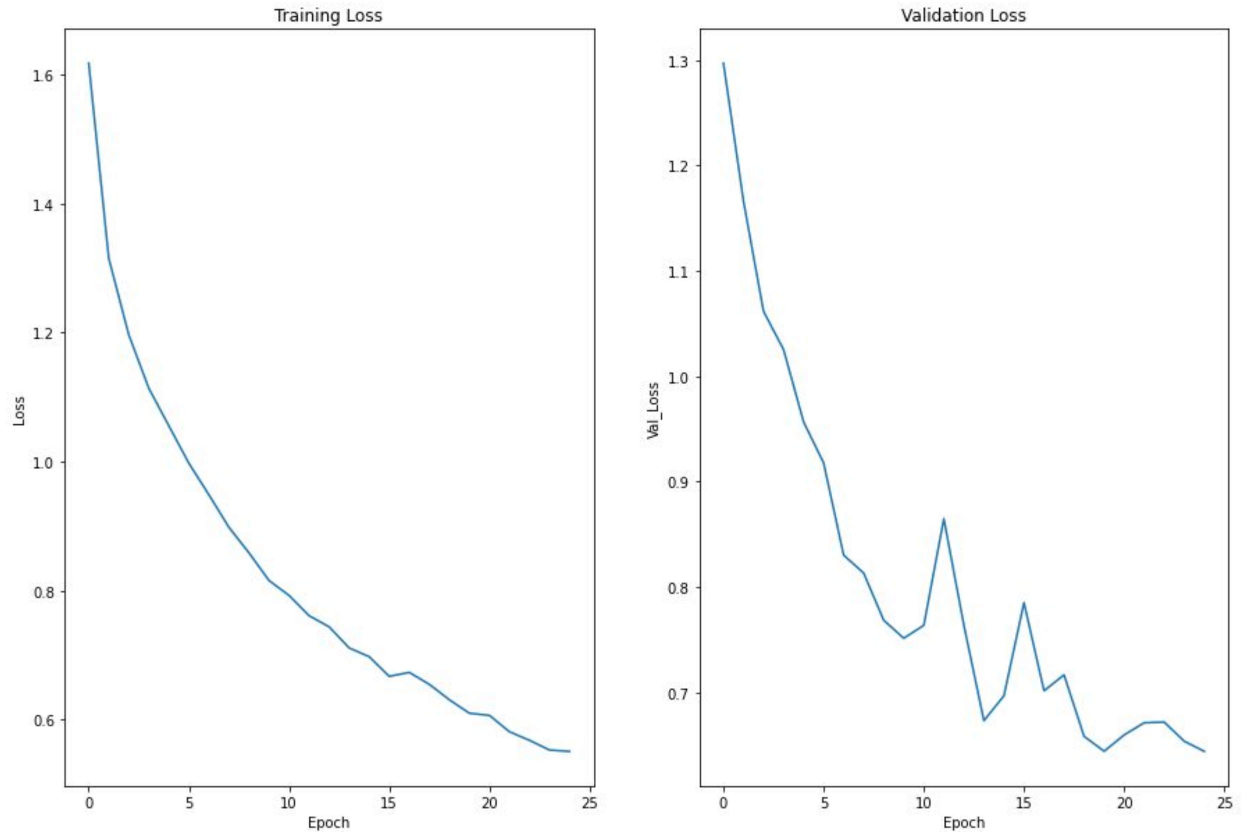
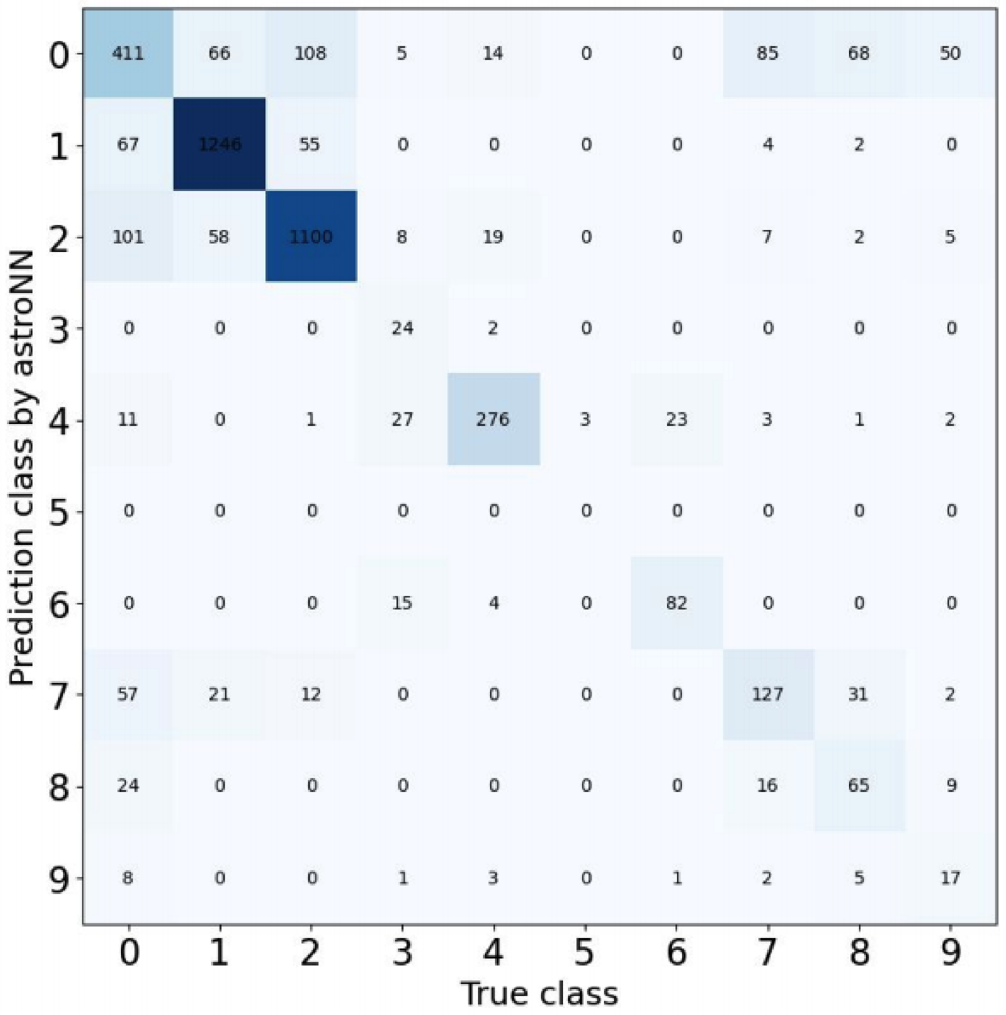


Figure 2: Training and Validation Loss Over 25 Epochs
On the left we observe a fairly smooth graph in which our loss decreases to approximately 0.5, while on the right we see a general decrease in loss over the epochs, albeit jumpy.

We also wanted to view our data in the form of a confusion matrix in figure 3 below. We used a built in astroNN function which allowed us to input our customized model and predictions into the confusion matrix. The initial confusion matrix compares the actual classification label (row side) to the model's predictions (column side). We notice that while the predictions on classes 0, 1, 2, 4 and 6 are relatively good given the data tested, the rest have poorer predictions. For instance, class 5 predicts incorrectly all three times. However this could be due to a small training set on class 5. This confusion makes sense as well because class 5 is "Disk, Edge-on, Boxy Bulge" while class 4 is "Disk, Edge-on, Rounded Bulge." There is a very slight distinction between boxy bulge and rounded bulge, so the confusion is understandable.



	0	1	2	3	4	5	6	7	8	9
0	411	66	108	5	14	0	0	85	68	50
1	67	1246	55	0	0	0	0	4	2	0
2	101	58	1100	8	19	0	0	7	2	5
3	0	0	0	24	2	0	0	0	0	0
4	11	0	1	27	276	3	23	3	1	2
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	15	4	0	82	0	0	0
7	57	21	12	0	0	0	0	127	31	2
8	24	0	0	0	0	0	0	16	65	9
9	8	0	0	1	3	0	1	2	5	17

Figure 3: Confusion Matrix of Predictions vs True Class Labels
The rows correspond to the prediction and the columns correspond to the true class. We can see that there is relatively low confusion on classes 0, 1, 2 and 4 and significantly higher confusion on classes 5 and 9.