Why our computer is better than you at differentiating galaxy morphologies

Aaron Desrochers,[1] Loïc Miara,[1]
[1]*McGill Physics Department,*

15 April 2022

ABSTRACT

In this paper, we use a Convolutional Neural Network (CNN) to build a Machine Learning (ML) algorithm that is able to classify images of different galaxies based on their morphology. To do this, we trained and tested our algorithm with the Galaxy10 DECals dataset, which can be found here. After tinkering with our model, we were able to achieve an accuracy of 65% with our ML algorithm when classifying the galaxy images into 10 distinct categories. After grouping similar galaxy morphologies together into broader categories and removing problematic morphologies, our algorithm was able to reach an accuracy of 91% when classifying the galaxy images into 3 categories: smooth galaxies, spiral galaxies and edge-on galaxies. We also determined that a higher image resolution did not improve the performance of the CNN, even with an additional convolution layer. We believe we have come close to the highest possible accuracy, which is limited by the subjectivity of galaxy classification itself. We see a real world application for our algorithm, seeing as software like "Zooniverse", from which the original labeling of the galaxies in this dataset is from, still rely on volunteers to manually classify galaxies.

Key words: Convolutional Neural Network – Galaxy Morphology – Image Classification

1 INTRODUCTION

Machine learning in astronomy and astrophysics is becoming increasingly useful to deal with large amounts of data. Essentially, ML is a blown-up curve fitting problem, where the number of fit parameters is much larger than for simple physical models (e.g.: $\sim 10^5$ in the proposed model). In this paper, we build and apply a CNN for the task of image classification. More specifically, we classify galaxy images according to their morphological type. Having a computer automatically classify images removes the arbitrariness of having it done by puny humans. We can then say that it standardizes the process of galaxy morphology classification.

Another goal of this exercise was to familiarize ourselves with machine learning libraries in Python. It has become very accessible through Tensorflow and Keras to code up machine learning algorithms, so this paper is a testament to its accessibility.

1.1 Convolutional Neural Networks

A CNN is function with a very high number of parameters that takes as input 2D images as arrays and returns a vector. Applied to image classification, each component of the output vector represents a confidence level in predicting the result associated with the index of the component. To build this function, we use convolution layers. Convolution refers to the process of taking a small box smaller than the size of the image (a filter with a kernel size) and going over all of the image pixels, and multiplying the RGB values of each pixels by a parameter in the model (a weight) [Brownlee (2017a)]. This boils down to matrix multiplication. We then sum over all the values of the filter and send the resulting value to the next layer of the network. Going over the whole input image like this gives the CNN a way to detect features smaller than the image, which is what we want to achieve for image classification. Each convolution layer is also accompanied by a max pooling layer, which, not unlike the convolution, moves a box smaller than the size of the image across the image and returns the highest pixel value. Once the CNN outputs something, we compare the resulting output to the desired output using training images, of which we know the type. We then compute the loss, which is a measure of how far away the model was from the desired output. Considering how wrong the model is, we then modify its parameters (the weights) so that the guess is now closer to the desired output. Repeating this for a large amount of data is referred to as "training". Once the model has been trained enough, we can use it to make predictions on data it has never seen before and quantify the performance of the model. In this paper, we will investigate how the pre-processing of the data affects the performance of the network, namely the number of morphological categories of galaxies we want to map our images to and the resolution of the images.

2 METHODS

2.1 The Model

The CNN we are using is taking in the images as $N \times N \times 3$ arrays where N is the sidelength of the image in pixels and the 3 represents the 3 color channels of the image. The original images are square with a sidelength of 256 pixels. We then have the liberty of resizing them to any sidelength smaller than the original size. The input layer takes in this image and does a first convolution and max pooling. We then have three hidden convolution layers and one hidden dense layer. The output layer has a number of neurons equal to the number

© 0000 The Authors

Layer	$N_{filters}$	Kernel size	Activation function
Convolution	32	3×3	ReLU
Max Pool	-	2×2	-
Convolution	64	3×3	Sigmoid
Max Pool	-	2×2	-
Convolution	64	3×3	Sigmoid
Max Pool	-	2×2	-
Dropout (2%)	-	-	-
Convolution	48	3×3	Sigmoid
Max Pool	-	2×2	-
Dense	64	-	Sigmoid
Dense	$N_{categories}$	-	Sigmoid

Table 1. The architecture of the model where the top row is the input layer and the bottom row is the output layer. The loss function is Sparse Categorical Crossentropy [Elijah Koech (2020)] and the optimizer is Adam [Brownlee (2017b)].

of categories we want to classify. Table 1 shows the architecture of the model in detail. Each model is then trained for a total number of 30 epochs, where an epoch is the number of times we feed the training dataset through the algorithm.

2.2 The Data and it's Preprocessing

The original data is a set of 17736 256×256 images with 3 color channels taken from the DESI Legacy Imaging Survey and compiled by Henry Leung and Jo Bovy from the University of Toronto [Bovy & Leung (2017)]. Table 2 Shows the distribution of each galaxy type in the original dataset. However, we notice that some galaxy types are way more prominent in the dataset than others. To account for this, we take the data from the categories which have a smaller number of images and create rotated duplicates of the images. We add the duplicate data until all the classes reach the highest number of images (2645). We also resize all of these images to a smaller image size. For the rest of this paper, if the sidelength of the image is unspecified, we assume that the image has been resized to 50×50 pixels.

After this process, we shuffle the order of the images in the dataset and preserve 10% of them for testing.

Category	N_{images}
Disturbed	1081
Merging	1853
Round Smooth	2645
In-between Round Smooth	2027
Cigar Shaped Smooth	334
Barred Spiral	2043
Unbarred Tight Spiral	1829
Unbarred Loose Spiral	2628
Edge-on without Bulge	1423
Edge-on with Bulge'	1873

Table 2. The distribution of galaxies of the original dataset

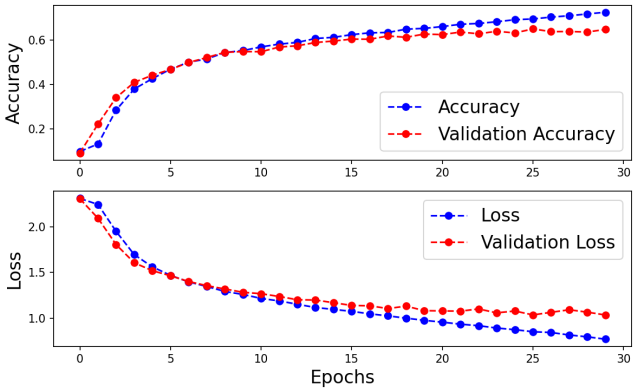

Figure 1. The accuracy, validation accuracy, loss and validation loss for the model as it trains over multiple epochs. We can see that the validation accuracy and validation loss flatten out around 20 epochs, demonstrating the limit of our model.

3 RESULTS

To get an efficient machine learning algorithm, it took a lot of trial an error, varying components of the system to have it finally produce proper results. In this section, we'll detail some of these changes we made to end up with our final models.

3.1 Original Model

When training the model to recognize all 10 galaxy morphologies, we end up with an accuracy of around 65%, as can be seen in Figure 1. We notice that the validation loss starts oscillating aroung the same value when we pass around 20 epochs of training, which indicates that the model reaches its predicting limit.

However, it is hard to blame the model for the low predicting power, as some categories are very much alike, especially when compressing the images. To illustrate this, Figure 2 shows an example of an instance where the model predicted that the presented galaxy was indeed a spiral one, but it got mixed up between two sub-categories of spiral galaxies.

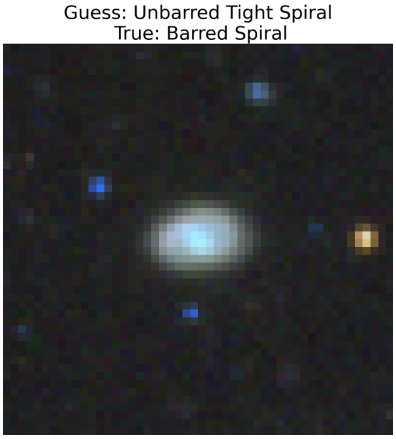

Figure 2. Example of a wrong guess made by the algorithm

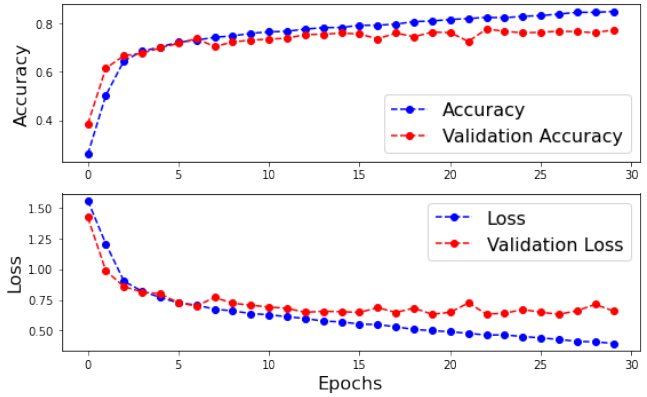

Figure 3. The accuracy, validation accuracy, loss and validation loss for the model as it trains over multiple epochs for the grouped morphologies. We can see that the validation accuracy rises quicker than when the galaxies weren't grouped and flattens out around 15 epochs, demonstrating the limit of our model.

3.2 Galaxy Grouping Model

In an attempt to give our algorithm a break and increase our prediction accuracy, we decided to group some of the similar galaxy types together. This meant that round smooth, in-between round smooth and cigar round smooth galaxies were all grouped together under the "Smooth Galaxy" label. Barred spirals, unbarred tight spirals and unbarred loose spirals were also grouped together as "Spiral Galaxies", and finally edge-on with bulge and edge-on without bulge both simply became "Edge-On Galaxies". The most prominent category is now "Spiral" (6500 images), so all other categories have their images rotated and duplicated until we reach this number of images across categories. With this new grouping, our algorithm was able to attain an accuracy of 77% with its predictions. The loss and accuracy curves can be seen in Figure 3. We also notice that with this grouping, the training is more efficient, as the accuracy and loss curves plateau out in less epochs.

	Precision (%)	N_{images}
Disturbed Galaxy	67	684
Merging Galaxy	83	660
Smooth Galaxy	71	632
Spiral Galaxy	74	639
Edge-On Galaxy	91	634
Macro Average	77	3249

Table 3. The accuracy of the machine learning algorithm with only 5 galaxy types; disturbed, merging, smooth, spiral and edge-on. The support images are the number of images kept from the original training dataset for testing the model.

	Precision (%)	N_{images}
Merging Galaxy	89	668
Smooth Galaxy	84	628
Spiral Galaxy	84	660
Edge-On Galaxy	94	643
Macro Average	88	2599

Table 4. The accuracy of the machine learning algorithm with only 4 galaxy types; merging, smooth, spiral and edge-on. We notice that the accuracy is best for edge-on galaxies, which is consistent with expectations as convolution is great at doing edge-detection.

3.3 Grouped Model Without Disturbed Galaxies

Our final optimization idea was to remove all galaxies labeled as "Disturbed Galaxies" from our dataset. The "Disturbed Galaxies" label seems to just be a catch-all term for all galaxies that don't fit into the other 9 categories. For this reason, it is difficult for the machine learning algorithm to properly find patterns in the different galaxies falling under this label, so in an attempt to increase its accuracy even more, we attempted running the model without including the disturbed galaxies.

We attempted with our grouped galaxies, meaning we now only had 4 different types of galaxies: merging galaxies, smooth galaxies, spiral galaxies and edge-on galaxies. With this new dataset, we were able to achieve a machine learning algorithm which correctly predicted the galaxy type 88% of the time. Table 4 shows the distribution of precision across morphologies in detail.

3.4 Increasing Pixel Density

To see if we could break this 88% accuracy barrier, we run a slightly modified version of our model with higher resolution images. Because the input data has more dimensions, we can add another convolutional layer to the network of Table 1. The added layer is identical to the last convolution layer. However, even with 100×100 sized images, we were still only able to reach that same accuracy of 88%. We then conclude that having higher resolution does not result in an improved performance from the network. However, we notice that most wrong guesses have to do with the merging galaxies type (see Figure A1 for examples). Indeed, galaxies can be both merging and have a definite morphology (e.g.: merging and spiral). However, the labels from the original data does not allow to be in two categories at the same time, hence, the model is forced to choose between two correct answers but only one of them is "truly" correct.

3.5 Guessing Only Three Classes

To try to solve this issue, we remove the merging galaxies images, similarly to how we removed the disturbed galaxies, in an attempt to improve accuracy. Previously, we had a noticeable 11% increase in predicting power when removing a category. However, removing the merging category only improved accuracy by 3%.

We notice that it is becoming increasingly harder to improve the predicting power of the network. We will hence investigate the original labeling of the data to see if there could have been a bias that intrinsically prevents us from improving accuracy by a high margin.

	Precision (%)	N_{images}
Smooth Galaxy	86	617
Spiral Galaxy	93	660
Edge-On Galaxy	93	672
Macro Average	91	1949

Table 5. The accuracy of the machine learning algorithm with only 3 galaxy types; smooth, spiral and edge-on. This is only marginally better than the accuracy of the model for 4 categories represented in Table 4.

$N_{categories}$	Precision (%)	N_{images}
10	65	26442
5	77	32499
4	88	26000
4 (100×100 pixels)	88	26000
3	91	19500

Table 6. Summary of the accuracies of our model with different training datasets and output categories. Recall that 10% of N_{images} is kept for testing, from which we obtain the precision.

4 DISCUSSION

As we increase the resolution of the images we fed to our machine learning algorithm, we still weren't able to increase our accuracy past 88%, which is due to multiple reasons. First off, in our dataset, each galaxy image can only be labeled as 1 morphology, but sometimes it needs to have multiple labels. For example, if you have 2 spiral galaxies next to each other, is this image of merging galaxies, or just of a spiral galaxy. Or even worse, what if you have a spiral galaxy next to a smooth galaxy, which one should the algorithm pick? This leads to the second reason why we think we can't pass this 88% barrier, which is that this galaxy labeling is subjective. Sometimes the galaxies are too small to properly categorize them, or sometimes they could fall into both spiral or edge-on galaxies depending on who you ask. Since there are no clear cut-offs between the 4 morphologies, there is an inherent subjectivity to classification which means that an algorithm would never be able to be close to 100% accurate. There were lots of instances where our algorithm guessed wrong, and we actually agreed with the guess of the algorithm over the actual provided label, further demonstrating why we hit this wall at around 90% accuracy. Finally, the last reason why we think we are restricted to this kind of accuracy is that the labeling for the images in our dataset was provided by the "Zooniverse" software, which is based on people classifying galaxies on a volunteer basis, then aggregates the results and picks the label based on the responses. Anyone can do this and on their website there aren't clear cut distinctions between the different morphologies. Indeed, the consensus across volunteers is only rarely greater than a significance level of 5% [Simmons et al. (2016)]. You can try this manual galaxy classification yourself on their website here, and you'll see how tough some of the distinctions can be, and how you could see the galaxy being part of multiple labels.

This then confirms an underlying human bias to the raw data, which intrinsically prevents our network from achieving image recognition that is on-par with other ML algorithms that deal with images with more obvious distinctions between them, like the MNIST handwritten digits datasets, for which CNNs are able to reach accuracies of $> 99\%$ [Gupta (2020)].

5 CONCLUSIONS

After constructing our original model, we experimented with grouping some galaxy morphologies together and with removing morphologies which may confuse the machine learning algorithm. With this experimentation, we were able to increase the accuracy of our algorithm from 65% (for 10 categories) to 91% (for only 3 categories).

Seeing as the labeling for our dataset was done entirely by volunteers over at Zooniverse and their manual classification is still up and running, we see a real application for our machine learning software in categorizing new galaxies much quicker than the previous model of needing volunteers. A reverse approach to the manual labeling of galaxies could be to let a CNN classify newly observed galaxies and then have volunteers agree or disagree with the proposed guess. This process would be even faster than the current manual labeling of the Zooniverse software and we believe it could be more accurate than human-biased classification.

Nevertheless, our model achieved satisfying accuracies, especially when grouping galaxies together, which was one of the objectives of this paper. We also report that the process of building and running ML algorithms was very accessible, and that the online documentation is more than enough to create and operate a CNN that is on-par with the proposed one.

ACKNOWLEDGEMENTS

We would like to thank B. Jovy and H. Leung for the easily accessible dataset, as well as A. Liu for the helpful conversations.

6 AUTHOR CONTRIBUTION STATEMENT

The script for the algorithm was developed fully in team working side by side. Most of the analysis was first discussed orally and L. Miara wrote slightly more than half the report while A. Desrochers cleaned-up and documented the code needed to run the algorithm.

DATA AVAILABILITY

The entire dataset of all the galaxies and their labels can be found here. The code used to create the machine learning algorithm can be found here.

REFERENCES

Bovy J., Leung H., 2017, Galaxy10 DECals Dataset, https://astronn.readthedocs.io/en/latest/galaxy10.html

Brownlee J., 2017b, Gentle Introduction to the Adam Optimization Algorithm for Deep Learning, https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Brownlee J., 2017a, How Do Convolutional Layers Work in Deep Learning Neural Networks?, https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/

Elijah Koech K., 2020, Cross-Entropy Loss Function, https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e#:~:text=Categorical%20cross%2Dentropy%20is%20used,%5D%20for%203%2Dclass%20problem.

Gupta J., 2020, Going beyond 99% — MNIST Handwritten Digits Recognition, https://towardsdatascience.com/going-beyond-99-mnist-handwritten-digits-recognition-cfff96337392
Simmons B. D., et al., 2016, Monthly Notices of the Royal Astronomical Society, 464, 4420

APPENDIX A: FIGURES

This paper has been typeset from a TEX/LATEX file prepared by the author.

Figure A1. 2 random samples from the test data when running the model for 100×100 pixels images. The guesses form the network as well as the true labels of the images are listed on top of each image. We notice that the wrong guesses are due to the algorithm guessing that the image displays merging galaxies, but the image is not labeled that way.