Evaluating Neural Network Models for Comet Identification

Juan Carlos Manjarres and Dr. Raymond Spiteri

April 2022

Abstract

The latest advancements in Neural Networks and Artificial Intelligence have allowed for the development of new ways of processing astronomical data. In this paper, we use astronomical data available from comets to create various Neural Network models, that we later compare based on their accuracy and efficiency. The Neural Network models are created using TensorFlow, Python and Keras, and the data is based on Category C comets, which is openly available thanks to NASA's Solar and Heliospheric Observatory (SOHO).

1 Introduction

Astronomy-related data has been growing exponentially for a number of years thanks to the rise of new telescopes, technology and image processing.

Thanks to the rise of Artificial Neural Networks (ANNs) and new image recognition techniques using Artificial Intelligence (AI), astronomers and computer scientists have been able to identify and create new mechanisms to categorize astronomical bodies. These mechanisms include the rise of AI projects such as StarcNet [10] and Morpheus [2], which identify star clusters and classify galaxies morphologically, respectively.

However, there is a lack of neural networks dedicated to detecting or identifying specific astronomical bodies such as black holes, comets, and even new stars; while the data to create these networks is already available and growing. And although there continue exist roadblocks such as the lack of documentation and the difficulty of processing such large datasets, there is a need for an approach that could be more computer-based.

Therefore, this paper will evaluate a variety of Artificial Neural Networks in regards to their accuracy for identifying Category C comets. These types of comets are notoriously faint and difficult to identify by humans, which strengthens the need for a more automated approach that can utilize existing data for predictions [7].

2 Related Work

As mentioned before, some work has been done in the area, specially in regards to the classification of spectroscopy data, and the creation of new tools such as astroML, which help to use the available datasets and existing machine learning algorithms for multiple applications, such as density and magnitude estimations, or classification of galaxies [14].

Additionally, there has been research that utilizes these neural network tools in astronomy, such as the application of Convolutional Neural Networks (CNNs) to estimate the redshift of galaxies [11] and the improvement of resolution in cosmological simulations thanks to AI [8]. In regards to astronomical image data, Morpheus has been detecting and characterizing galaxies since its inception, utilizing Python and TensorFlow for Deep Learning [2]; while StarcNet has been useful for identifying star clusters in galaxies, using a CNN [10].

Moreover, in terms of smaller astronomical bodies, there has been some work done identifying potential asteroids that could impact the Earth, thanks to the development of an ANN by Hefele et al [3].

Outside of the astronomy field there has been some research on Multi-input CNNs, which are essential for this problem, as the data used requires multiple images per comet. The research on Multi-input CNNs includes varied applications such as classification of microscopy images for studying cell biology, and the grading of flowers for managing greenhouses. These applications use different approaches to multi-input CNNs: while the microscopy images use a combination of pooling layers and Multiple Instance Learning (MIL) [5], the grading of flowers use a more common three-input model that utilizes a traditional CNN [12]. These approaches to Multi-input CNNs were useful as a basis to the model used in this paper.

3 Dataset

The dataset for this problem is composed of multiple images per comet, which were gathered from the NASA's Solar and Heliospheric Observatory (SOHO)[7].

The training dataset consists of about 2000 comets, and each comet contains at least one .FITS image. Every .FITS image has a size 1024x1024, and by standard it is only single-channel, with each pixel representing brightness –i.e., a gray-scale image.

However, because of the faintness of these Category C comets, it is necessary to use at least 5 images to avoid problems with instrument noise, as it could lead to misidentified comets. Therefore, the models for this research only use comets with at least 5 images.

This dataset also included a ground truth file, which was used for the training, as it specifies the location of the comet on each image, as a pixel coordinate.

This truth data was loaded in Python [13], using the existing libraries for opening text data. Afterwards, the data was stored in a Python dictionary for easy access, in which each key represented the comet number. This comet number was then mapped to another dictionary, which contained the filename of each image as well as the pixel coordinates of the location of the comet.

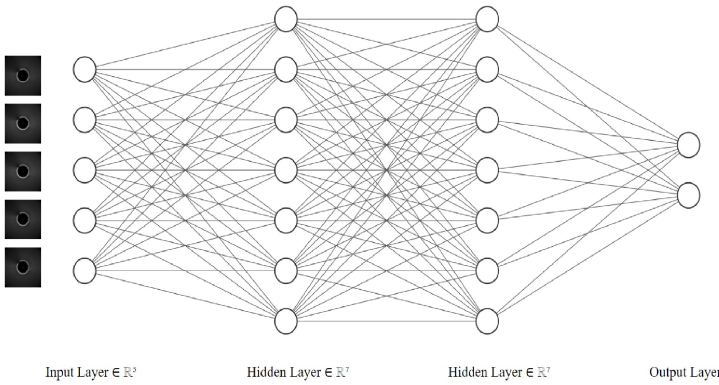

Figure 1: Neural Network architecture for the Model, which consists of 4 layers, with one 5-input layer and one 2-output layer. The hidden layers are consisted of 7 nodes each.

4 The Main Model

As specified on the Related Work section, we had to envision a Neural Network that allows for the input of multiple images; therefore, Multi-input CNNs are the strongest option, as they are mainly used in similar problems, such as the Flower Grading example mentioned before.

With the help of Python [13], TensorFlow [6] and various other libraries [9] [1] it was possible to create an initial CNN that was trained on the images provided.

Because of the nature of the problem, which would give as a solution a tuple with the coordinates of the comet, the model was based on Regression techniques designed for TensorFlow, although more sophisticated techniques can be explored in future research, such as Bayesian regression, which are used in multiple examples on astroML [4].

To avoid problems with the input variable size, the Neural Network was created with 5 input nodes, that would allow for the detection of the comet per NASA's specifications.

These input nodes take one .FITS image per node, which were converted to TensorFlow tensors using the tf-fits package [9]. This allows for easier handling of the .FITS images.

Additionally, the network is composed of 2 distinct Dense hidden layers, each with 7 nodes. The quantity of the layers and the nodes were chosen as related examples in astroML use this configuration; but other configurations can be tested for future research.

One main drawback to the model is the storage and computing requirements; hence it was trained with 100 comets instead of the full set, as the machine used had some limitations.

5 Results

Based on the main network defined above, we utilized three main loss functions, and evaluated them based on their performance for this problem. The performance was gathered through 70 epochs, to evaluate their accuracy and loss with the dataset. Training past 70 epochs did not improve any particular results, hence these are the best networks

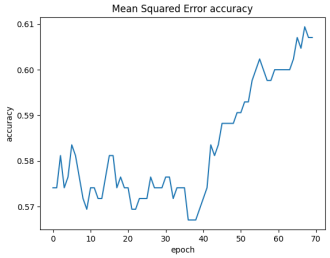

Figure 2: The accuracy of the Mean Squared Error function.

The network that utilized the Mean Squared Error loss function gave an accuracy of about 60% after the 70 epochs were done. The loss also improved dramatically throughout the epochs, although it was still higher than anticipated. This could be due to a number of factors, but an important one to highlight is the sheer size per image, and the Mean Squared Loss having usually bigger loss sizes due to its squared formula.

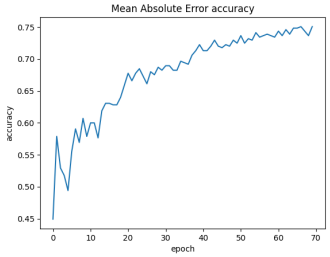

Figure 3: The accuracy of the Mean Absolute Error function.

On the other hand, the network with the Mean Absolute Error loss function gave a much lower loss at the end of the 70 epochs, while it also improved throughout each epoch (from 1252.29 on the first epoch, to 317.30 on the last one). The lower loss compared to the Mean Squared Error was expected, because of the difference in the formulas to calculate them [6]; however, the accuracy also improved to about 75%, which is significant in comparison with the Mean Squared Error neural network.

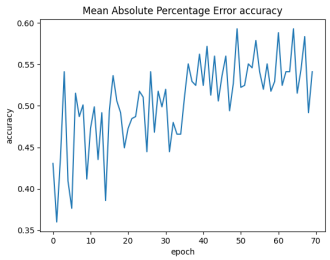

Figure 4: The accuracy of the Mean Absolute Percentage Error function.

Lastly, the Mean Absolute Percentage Error network did not perform very well. Although the loss was not very big (with a last epoch loss of 211), its accuracy was very unreliable, as it bounced between 35% and 60% between epochs. Additionally, while all the other neural networks had major improvements on their accuracy throughout epochs, this network was the exception. Hence, this loss function is not an ideal candidate for this particular problem.

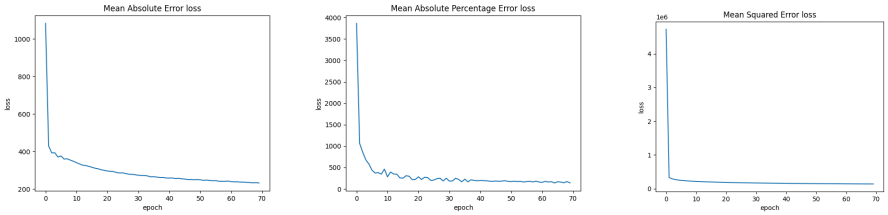

Figure 5: The losses for all three networks tested.

6 Conclusion

We conclude that a Neural Network is a viable way for identifying Category C comets. Additionally, through testing, it was possible to compare these networks, and based on the data gathered through the accuracy and the losses, it is suggested that a network with a Mean Absolute Error loss function can be more effective towards building an accurate model for this problem.

Moreover, there exist multiple avenues for further exploration on this topic, starting with the application of other types of networks, such as the MIL network utilized for microscopy [5], or networks in the TensorFlow framework, such as Recurrent Neural Networks [6]; the use of different nodes and techniques, such as Flatten nodes and Concatenation; and the use of different activation functions.

References

[1] Astropy Collaboration et al. "Astropy: A community Python package for astronomy". In: 558, A33 (Oct. 2013), A33. DOI: 10.1051/0004-6361/201322068. arXiv: 1307.6212 [astro-ph.IM].

[2] Ryan Hausen and Brant E Robertson. "Morpheus: A Deep Learning Framework for the Pixel-level Analysis of Astronomical Image Data". eng. In: *The Astrophysical journal. Supplement series* 248.1 (2020), p. 20. ISSN: 0067-0049.

[3] John D Hefele, Francesco Bortolussi, and Simon Portegies Zwart. "Identifying Earth-impacting asteroids using an artificial neural network". eng. In: *Astronomy and astrophysics (Berlin)* 634 (2020), A45. ISSN: 0004-6361.

[4] Željko Ivezić et al. *Statistics, data mining, and machine learning in astronomy: a practical Python guide for the analysis of survey data*. eng. Updated edition. Princeton series in modern observational astronomy. 2020. ISBN: 9780691198309.

[5] Oren Z Kraus, Jimmy Lei Ba, and Brendan J Frey. "Classifying and segmenting microscopy images with deep multiple instance learning". eng. In: *Bioinformatics* 32.12 (2016), pp. i52–i59. ISSN: 1367-4803.

[6] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[7] *NASA SOHO Comet Search*. https://www.nasa.gov/nasa-soho-comet-search. Accessed: 2022-08-04.

[8] Yueying Ni et al. "AI-assisted superresolution cosmological simulations – II. Halo substructures, velocities, and higher order statistics". eng. In: *Monthly notices of the Royal Astronomical Society* 507.1 (2021), pp. 1021–1033. ISSN: 0035-8711.

[9] William J. Pearson. *TF-fits*. URL: https://pypi.org/project/tf-fits/.

[10] Gustavo Pérez et al. "STARCNET: Machine Learning for Star Cluster Identification". eng. In: *The Astrophysical journal* 907.2 (2021), p. 100. ISSN: 0004-637X.

[11] Radamanthys Stivaktakis et al. "Convolutional Neural Networks for Spectroscopic Redshift Estimation on Euclid Data". eng. In: *IEEE transactions on big data* 6.3 (2020), pp. 460–476. ISSN: 2332-7790.

[12] Yu Sun et al. "Multi-Input Convolutional Neural Network for Flower Grading". eng. In: *Journal of electrical and computer engineering* 2017 (2017), pp. 1–8. ISSN: 2090-0147.

[13] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[14] Jacob Vanderplas et al. "Introduction to astroML: Machine learning for astrophysics". eng.

In: *Proceedings - 2012 Conference on Intelligent Data Understanding, CIDU 2012*. 2012, pp. 47–54. ISBN: 146734625X.