Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Report

Compression of Astronomy Pictures using Neural Networks

Vincent Stettler & Mohammed Ajil

Contents

1 Introduction

Missions by space agencies often involve sending spacecrafts to space in order to study objects of the solar system. On October 15, 1997, for example, an unmanned spacecraft called *Cassini* was launched to study the planet Saturn. After entering orbit on July 1, 2004, it spent 13 years studying the planet and its system. The *Imaging Science Subsystem* (ISS), using two CCDs, took thousands of images of Saturn, its rings, and its moons. Since both the storage capacity and the network bandwidth were limited, it was necessary to use data compression in order to increase the number of images that could be taken and sent back to earth.

Missions such as the Cassini mission currently use lossy image compression algorithms like JPEG. These algorithms are designed to yield good results across a large variety of image domains, and are thus not adapted to the specific domain of space exploration.

In recent years, neural networks have shown successes across many different applications. The question whether and how they can be used for the task of image compression has thus received growing interest.

One natural idea would be to use autoencoders [2], which are trained to optimize the reconstruction loss while having a bottleneck in the architecture. One drawback of this approach however is the fixed compression rate that is given by the dimensionality of the bottleneck layer. To allow for variable-rate compression, Toderici et al. [3] [4] experiment with *stacked* autoencoders, where multiple autoencoders are chained together and each stage is trained to reconstruct the residual of the previous stage. Baig et al. [1] investigate different approaches for chaining succesive stages of such architectures.

In this work we implement different models discribed by Baig et al. [1] in tensorflow and see how they perform on images of the DECam DR4 dataset (`http://legacysurvey.org/decamls/`) We first discuss this dataset and then describe the models that were used. Finally we introduce the experiments that were done and present the results.

2 Dataset

For the training and evaluation of the models we used three datasets. These were DECam DR4, the Cassini Dataset, and ImageNet. ImageNet is a well known dataset, therefore we will not address it further. In the following we will shortly describe the other two datasets and describe how which transformations we used before feeding them into the models.

2.1 DECam DR4

DECam stands for Dark Energy Camera, DR4 for data release 4. The data we used is available at[1]. In the repository at[2] there is a script that downloads the data release. The data release 4 is part of the "DECam Legacy Survey of the SDSS Equatorial Sky", this survey aims to probe the dynamics of the expansion of the Universe and the growth of large scale structure. We chose this dataset because of its completeness and the cleanness of the data that is provided to the public. The data is provided in the FITS data format. In Python we can use `astropy`[3] to read this dataformat. The images consist of one channel, and the values per pixel range from 0 to 2^{16}. Before feeding the images into the model we rescale them to a range of $[-1, 1]$ and flip the pixel values.

why did we do that again?

Figure 1: DECam image original

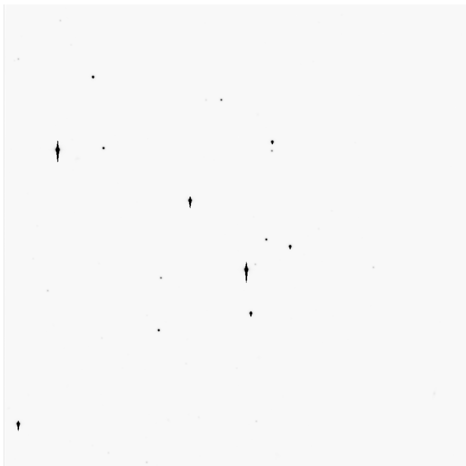

Figure 2: DECam image rescaled

[1] `http://legacysurvey.org/dr4/`
[2] `https://gitlab.com/mohammed4/data-science-lab`
[3] `http://www.astropy.org`

2.2 Cassini Dataset

As mentioned in the introduction, Cassini is a spacecraft launched to study the planet Saturn and its system, including its rings and natural satellites.[4] The filtered images as we used them for training the model can be obtained from here[5] The are the wide angle images taken of saturn. These images are already in the JPEG format, therefore the pixel values are between 0 and 255. Before feeding them into the model we again rescaled them to the scale $[-1, 1]$. Since we did not flip the pixel values, the rescaled image looks the same, when saved as a PNG. Below there is again an example.

Figure 3: Cassini image

Since these images are encoded in JPEG we can use `scipy` to easily read them into an array.

[4] https://en.wikipedia.org/wiki/Cassini-Huygens

[5] https://saturn.jpl.nasa.gov/galleries/raw-images?order=earth_date+desc&per_page=50&page=0&min_distance=1000000&max_distance=5000000&targets%5B%5D=SATURN&cameras%5B%5D=ISSWA

3 Models

4 Experiments

5 Results

6 Conclusion

References

[1] M. H. Baig, V. Koltun, and L. Torresani. Learning to inpaint for image compression. *CoRR*, abs/1709.08855, 2017.

[2] A. Krizhevsky and G. E. Hinton. Using very deep autoencoders for content-based image retrieval. In *ESANN 2011, 19th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 27-29, 2011, Proceedings*, 2011.

[3] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar. Variable rate image compression with recurrent neural networks. *CoRR*, abs/1511.06085, 2015.

[4] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell. Full resolution image compression with recurrent neural networks. *CoRR*, abs/1608.05148, 2016.