

# Reinforcement Learning for Astronomical Image Processing

Justin Wilmes

University of Colorado, Colorado Springs (UCCS)  
1420 Austin Bluffs Pkwy  
Colorado Springs, Colorado, 80918

## Abstract

Image enhancement is a necessary aspect of image capture, which machine learning has in recent years greatly assisted (Fang, Pang, and Yi 2020). In this paper we study reinforcement learning with image enhancement with a focus on astronomical photos; in doing so, we will investigate several techniques that have been attempted in other domains in recent years, and test a convolutional architecture using double deep-Q learning. We find most results are good, with much of the variance coming from the quality of the images in the dataset.

## Introduction

While humanity has looked to the skies for as long as it has existed, our current knowledge of the universe outside of earth is a relatively recent phenomena. Photography in particular has only been applied to astronomy since the late 1800s (Gill 1887), although once applied, it greatly improved the knowledge gained and shared from astronomy. An article from 1923 explains that longer camera exposures made visible “objects too faint to be seen with the eye even when aided by the most powerful telescope” (Stokley Jr 1923). The ability of modern cameras to record faint detail is what enables scientific advancement in cosmology and astrophysics.

However, modern photography is more complicated than in the past - images need to be processed, which may involve a variety of steps using non-learning algorithms such as debayering (also known as demosaicing) (Kwan et al. 2018), noise-reduction (Seybold et al. 2014), high-dynamic-range (Gao et al. 2020), contrast enhancement, and overexposure correction (DONG 2011). These traditional algorithms are very powerful, but may require manual input to produce a photograph that is aesthetic, or useful. Recent research using machine learning have shown that it is possible to improve upon these algorithms, even in their basic functionality, with some algorithms acting as replacements and others as additional components to existing algorithms (Fang, Pang, and Yi 2020).



Figure 1: Example of activities available when manually photo editing. Shown is Sony Photo Editor.

## Background

Before examining what reinforcement learning has been doing, it is good to understand what types of editing may happen with a photograph. Someone using a photo editor may, for example, adjust the contrast and brightness of an image (see figure 1). There are usually more complex features available, such as adjusting the brightness mapping of the photo as a whole, or even of each color in the photo - known as a Tone Curve or Light Curve.

One may adjust these features to produce a photo that has adequate brightness, shows a full dynamics range (i.e., the shadows are dark and the highlights bright), has proper color balance, and does not show excessive noise. Alternatively, a machine might learn how to use the parameters to produce as good or better a photo, or modify the photo in another fashion that accomplishes the same goals. The greatest difficulty with these edits is ensuring that no information is lost in the image - a problem exacerbated when the image is intended for scientific use, as is common in astronomical photos, as seen in figure 2.

## Related Work

Several reinforcement algorithms examined for this problem were ReLLIE (Zhang et al. 2021), DeepExposure (Yu et al. 2018), PixelRL (Furuta, Inoue, and Yamasaki 2019), and RSE-RL (Bajaj, Yang, and Wang 2023).

**ReLLIE** (Zhang et al. 2021) modifies photos by applying various light curves, which may be high-order polynomials,

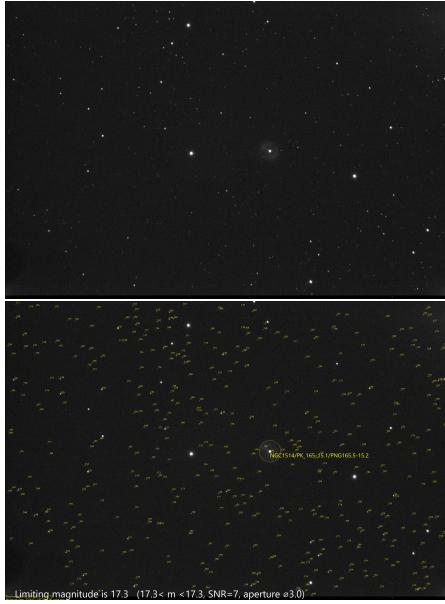


Figure 2: An example of the type of detail present in astronomical photographs. Photos by the author.

which the machine learns to either select from or build. This is similar to having a person edit a photo, since no human will try to edit each pixel individually; the process used in this method is fairly understandable, at least so far as the process that edited a particular photo. A similar method is explored in a paper titled “Unpaired Image Enhancement Featuring Reinforcement-Learning-Controlled Image Editing Software” (Kosugi and Yamasaki 2020), with the caveat that the focus is improving images produced by a GAN rather than using reinforcement learning as the primary editing tool. The work done by Zhang et al. served as the inspiration for the reward function and action space used in this project.

**DeepExposure** (Yu et al. 2018) operates similarly to high-dynamic-range - several images are taken of an object at various brightnesses, and then an algorithm selects the best part of each photo and splices them together to build a composite image. However, where high-dynamic-range may use a preset exposure (image brightness) values and analytical algorithm to splice images, DeepExposure uses intelligent selection of exposures and splicing. A somewhat similar process is used in a paper “Image Enhancement Using Adaptive RegionGuided Multi-Step Exposure Fusion Based on Reinforcement Learning” (Xi et al. 2023) - an image is encoded through one neural network, and then split and decoded through several different neural networks - and the outputs spliced - to artificially produce a similar effect.

**PixelRL** (Furuta, Inoue, and Yamasaki 2019) is perhaps the most computationally-challenging algorithm investigated - it utilizes an agent at each pixel (for a 9 mega-pixel image, this equates to 9 million agents) which modifies its pixel value based on the actions and status of the agents and

pixels around it, thus working at a lower level than other reinforcement learning methods. The applications of this algorithms are fairly broad and may include denoising, high-dynamic-range, and other operations.

**RSE-RL** (Recursive Self Enhancing Reinforcement Learning) (Bajaj, Yang, and Wang 2023) operates by recursively (hence the name) improving an image through a series of auto-encoders, the goal being to produce fewer artifacts in the improved image than are often found in other learning algorithms (such as guided neural networks).

## Methodology

### Datasets

Training has used two datasets for this training - one created by the author, and one a small set of sample images from the European Space Agency (ESA 2019). It was found in initial training that there was greater consistency in the images from the home-made dataset, so this was used for most of the training. It contains 57 monochrome images of varying subjects, brightness, and background noise. While this number is too small to effectively train a neural network on, the process for reinforcement learning - especially if the reward can be determined by a function, and the state space by the image’s histogram, rather than the image itself - makes this problem solveable.

### Reward Function

Based on the work of Zhang et al. (2021), the loss function used in this experiment is made of four components - three based on the mentioned paper, and an additional element to reward expansion of the histogram, since most of the images being manipulated have a fairly narrow range of values at the start. These can be seen in equations 1, 2, 3, and 4. The total reward is calculated according to equation 5. Weights are applied to the loss equations to allow tuning of the environment. Since the images being manipulated are monochrome, there is no need for further losses or rewards regarding color balance.

$$L_{spa} = \frac{1}{K} \sum_{i=1}^K \sum_{j \in \Omega(i)} (|Y_i - Y_j| - |I_i - I_j|)^2 \quad (1)$$

$$L_{exp} = \frac{1}{M} \sum_{k=1}^M |Y_m - E| \quad (2)$$

$$L_{tvA} = \frac{1}{N} \sum_{t=1}^N N \sum_{c \in \epsilon} (|\nabla_x A_t^c| + |\nabla_y A_t^c|) \quad (3)$$

$$R_h = \frac{\text{sum}(h)}{\text{max}(h)} - 1 \quad (4)$$

$$R = W_h * R_h - W_s * L_{spa} - W_e * L_{exp} - W_t * L_{tvA} \quad (5)$$

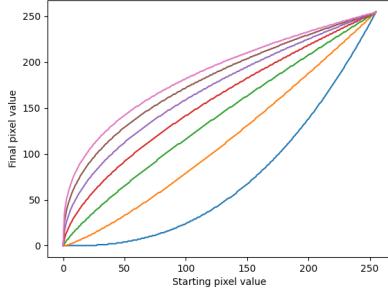


Figure 3: Curves the machine can pick from. These make up 7 of the 9 decisions possible for the machine to make.

## Actions

The action space for learning initially focused on using a series of disparate curves applied to the image, such as a logarithmic function, square-root, cubic, and others; however, this led to insufficient levels of adjustment for each curve applied. Later work focused on what are commonly known as  $\gamma$  correction curves, the equation for which can be seen in equation 6, where  $\gamma$  is the user-selected curve parameter,  $O$  is the output,  $I$  is an array corresponding to the possible values in the image histogram, and  $c$  is a constant related to the maximum possible value a pixel can have (usually 255). The actor is limited to a user-selected number of actions for each image (nominally 5).

$$O = c * (I/c)^\gamma \quad (6)$$

Several values of  $\gamma$  were chosen as actions the machine could take (see figure 3), along with two other functions - one which allows the machine to "do nothing", and one which attempts to map pixel values so that the histogram is as even as possible (this is known as "normalizing" the image). In all, this makes for 9 decisions in the decision space.

It should be noted that due to the non-linearity of these curves, there is a large difference in the image appearance when choosing between them. From lowest curve to the highest these map to decisions 1 through 7 (the decision space starts with a 0, which is the "do nothing" decision, and ends with 8, or normalize).

The other actions mentioned here - "do nothing" and normalize - allow the actor to either not modify the image (i.e. finish modifications) or use an algorithm that attempts to equalize the number of pixels at each intensity level in the histogram. While it seems as though this latter action should accomplish all editing needed, so that the actor would choose it and then do nothing for the remainder of the allowed actions for an image - however, in practice, this algorithm is unable to fully adjust the image by itself; while there were a few cases where the actor selected it and then nothing else (or "do nothing"), in most cases it would select either a series of curves, or do a curve, then normalize, then select a couple more curves.

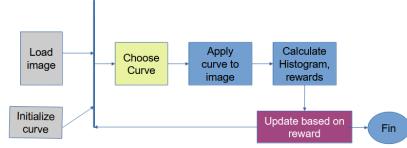


Figure 4: Example of the flow during a single episode. This process would be done many times during an epoch.

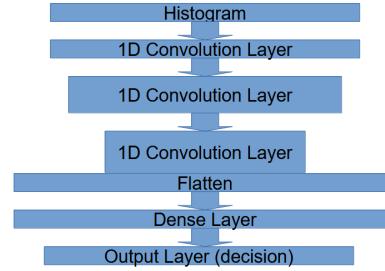


Figure 5: Neural network architecture used in this project.

## Architecture

An example of the process during a single episode is shown in figure 4.

Each training episode executes until either the reward has reached its maximum allowed value, or 5 iterations have been done (multiplying the curves together eventually leads to numerical issues, so the allowed iterations is limited to account for this limitation in the environment at the time of writing). Training in this situation used Double Deep-Q Learning, which produced acceptable results. The structure of the convolutional neural network used is shown in figure 5.

## Results

A pair of convolutional networks was used with Double Deep-Q Learning. Training results are shown in figure 6.

Figure 7 provides an example of what a successful image manipulation produces in the histogram - this is a desirable end-state for the model.

Figure 8 examines how training progressed over time. Because the rewards were predominantly negative, it was difficult to determine when the machine was finished learning. In addition to the CNN architecture used here, a simple dense network and an encoder were also tested - though these tended to stagnate after some point and failed to produce acceptable results.

It was found in testing that having a fairly large weight for the histogram spread (equation 4) led to best results, leading the average reward to average around 0.

## Difficulties Encountered

Unstability in convergence was the primary difficulty encountered in doing this project. Factors that led to this initially included situations where the reward for a blank image

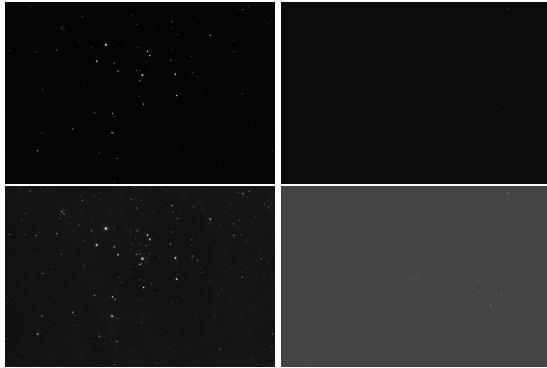


Figure 6: Example input (top) and output (bottom) images. It was noted that learning produced better results when the images contained greater variance and detail to begin with - there was some instability with dim, low-contrast images that could lead to good results or poor results.

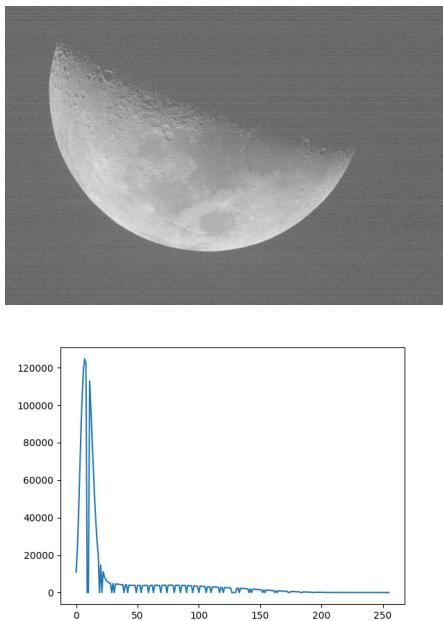


Figure 7: Example of a histogram and the corresponding image. The x-axis is the pixel value and the y-axis is the number of pixels with that value.

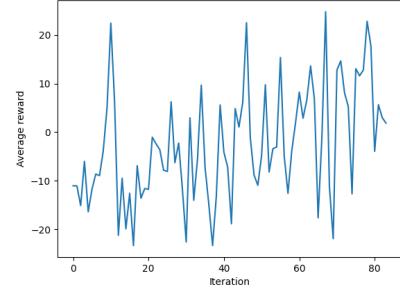


Figure 8: Training reward example. Moving average is calculated over each 100 episodes of training, more than 8000 episodes are shown.

was greater than for a well-balanced image. Additionally, the image set used for training started from images that were somewhat poor - although this was somewhat on purpose, it made it difficult for learning to overcome the many negative rewards, where choosing one action in a similar state may cause widely different results for different images. As the machine modified each image, it would produce different results for the same action - thus it ended up chasing a moving target, and if the random selection put too many of them in the same order, would lead it astray. These problems were helped by increasing the weight of the positive histogram reward,  $W_h$ , though not fully eliminated. After training, a the actor was tested on 49 images, which produced 32 successfully modified images, or about a 65% success rate.

Additional limitations include that the open library used to modify the images - Open CV (Bradski 2000) - does not work well with 16-bit images (most of the dataset is 16-bit images), which meant images were downsampled to 8-bit images, thus some detail was lost not due to actions taken by the actor. Because of this, order matters in how actions are taken, which the current architecture does a poor job of capturing - making it surprising that the brief attempts to use an encoder led to worse results; attempts to produce better results were not possible in the allocated time, and so were forgone in favor of optimizing the reward function.

## Conclusion

In this paper, we have applied reinforcement learning to edit astronomical photos. Example uses of this method would be to allow the immediate visualization of images that are taken as the come out of an astrograph, rather than needing processing in order to be useful. Ultimately the results were mixed, but a sufficient learning was accomplished to prove the environment is learnable and this method can be applied to astrophoto improvement.

## References

- Bajaj, C.; Yang, Y.; and Wang, Y. 2023. Reinforcement learning of self-enhancing camera image and signal processing. In *Advances in Data-driven Computing and Intelligent Systems*.

gent Systems: Selected Papers from ADCIS 2022, Volume 2. Springer. 281–303.

Bradski, G. 2000. The opencv library. *Dr. Dobb's Journal of Software Tools*.

DONG, G. 2011. A collection of digital photo editing methods.

ESA. 2019. Datasets for education and for fun.

Fang, W.; Pang, L.; and Yi, W. 2020. Survey on the application of deep reinforcement learning in image processing. *Journal on Artificial Intelligence* 2(1):39–58.

Furuta, R.; Inoue, N.; and Yamasaki, T. 2019. Pixelrl: Fully convolutional network with reinforcement learning for image processing. *IEEE Transactions on Multimedia* 22(7):1704–1719.

Gao, J.; Hung, S. R.; Velarde, R.; and Sun, G. 2020. Scene metering and exposure control for enhancing high dynamic range imaging.

Gill, D. 1887. The applications of photography in astronomy. *The Observatory*, Vol. 10, p. 267-272 (1887) 10:267–272.

Kosugi, S., and Yamasaki, T. 2020. Unpaired image enhancement featuring reinforcement-learning-controlled image editing software. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11296–11303.

Kwan, C.; Chou, B.; Kwan, L.-Y. M.; Larkin, J.; Ayhan, B.; Bell III, J. F.; and Kerner, H. 2018. Demosaicing enhancement using pixel-level fusion. *Signal, Image and Video Processing* 12(4):749–756.

Seybold, T.; Cakmak, Ö.; Keimel, C.; and Stechele, W. 2014. Noise characteristics of a single sensor camera in digital color image processing. In *Color and imaging conference*, volume 2014, 53–58. Society for Imaging Science and Technology.

Stokley Jr, J. 1923. Astronomy and photography. *Popular Astronomy*, Vol. 31, p. 373 31:373.

Xi, R.; Ma, T.; Chen, X.; Lyu, J.; Yang, J.; Sun, K.; and Zhang, Y. 2023. Image enhancement using adaptive region-guided multi-step exposure fusion based on reinforcement learning. *IEEE Access* 11:31686–31698.

Yu, R.; Liu, W.; Zhang, Y.; Qu, Z.; Zhao, D.; and Zhang, B. 2018. Deepexposure: Learning to expose photos with asynchronously reinforced adversarial learning. *Advances in Neural Information Processing Systems* 31.

Zhang, R.; Guo, L.; Huang, S.; and Wen, B. 2021. Relie: Deep reinforcement learning for customized low-light image enhancement. In *Proceedings of the 29th ACM international conference on multimedia*, 2429–2437.