International Journal of Image and Graphics

Reinforced Contrast Adaptation

H.R.Tizhoosh, G.W.Taylor

PAMI Research Group
Department of Systems Design Engineering
University of Waterloo
Waterloo, Ontario, N2L 3G1, Canada
tizhoosh@uwaterloo.ca, gwtaylor@engmail.uwaterloo.ca

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Traditional image enhancement algorithms do not account for the subjective evaluation of human operators. Every observer has a different opinion of an ideally enhanced image. Automated Techniques for obtaining a subjectively ideal image enhancement are desirable, but currently do not exist. In this paper, we demonstrate that Reinforcement Learning is a potential method for solving this problem. We have developed an agent that uses the Q-learning algorithm. The agent modifies contrast of an image with a simple linear point transformation based on the histogram of the image and feedback it receives from human observers. The results of several testing sessions have indicated that the agent performs well within a limited number of iterations.

Keywords: Reinforcement Learning; Q-Learning; Image Enhancement; Contrast; Subjectivity.

1. Introduction

Images are used every day to communicate information. The progression of digital images has not only changed the way people work with images, it has also increased society's dependence on high quality images. Low quality digital images can seriously affect the way an image is interpreted, introducing errors in analysis or eliminating the image's usefulness in a industrial or medical environment (1; 3; 5; 6; 10; 12; 16; 2; 9; 8).

The fundamental problem of image enhancement is that every person judges the quality of an image differently. When manipulating and enhancing images, there is not one ideal outcome that will satisfy everyone as every person has a different subjective perception. This requires that each observer spend the time altering image parameters, which can be time consuming and repetitive (7; 11; 14; 17; 21; 19).

While many software techniques for modifying images exist, methods to learn the subjective evaluation of a user and provide intelligent control for image processing are nonexistent. Ultimately, a method of determining the subjective image quality for a particular person is desirable. Determining such a technique could allow for the direct application of subjective image quality to future enhancement of digital images.

We introduce reinforcement learning as a technique for the automation of subjective image enhancement. Though it is continuing to prove successful in many applications, reinforcement learning has never before been used for this purpose. Using reinforcement learning for subjective image enhancement could initiate the development of personalized (observer-dependent) image enhancement profiles for the digital image community.

The motivation for this paper is that there are no advancements that combine the concepts of reinforcement learning and subject image enhancement. These advancements would apply to a wide variety of applications. Two main applications of this technology are medical imaging and industrial quality control.

The overall goal of this paper is to study the effectiveness of reinforcement learning for subjective image enhancement. Due to the broadness of both reinforcement learning and image enhancement, the scope of this project will be narrowed to deal with a subset of the issue. This paper will specifically deal with the study of using one particular type of reinforcement learning, namely Q-learning, for enhancing image contrast/brightness.

The paper is organized as follows: Sections 2 and 3 provide a background on reinforcement learning and image enhancement. Section 4 outlines our methodology for reinforced image enhancement. Section 5 describes the implementation and results of testing our algorithm. Finally, Sections 6 and 7 provide conclusions and future considerations.

2. Reinforcement Learning

Reinforcement learning is based on the idea that an agent learns by interacting with its environment (4; 15; 18; 22; 23). It allows software agents to automatically determine the ideal behavior within a specific context that maximizes performance. Several components constitute the general idea behind reinforcement learning. The agent, which is the decision maker of the process, attempts an action that is recognized by the environment. The agent receives from its environment a reward or punishment depending on the action taken. The agent also receives information concerning the state of the environment. The agent acquires knowledge of the actions that generate rewards and punishments and eventually learns to perform the actions that are the most rewarding in order to meet a certain goal relating to the state of the environment (Figure 1).

3. Image Enhancement

Image enhancement is the notion of modifying an image in order to acquire an optimal image quality. The quality of digital images depends on many factors (1; 5;

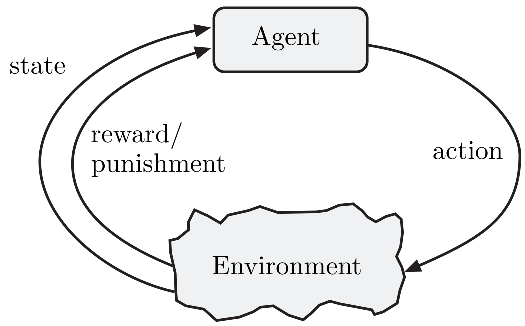

Fig. 1. Reinforcement learning block diagram

16; 2). Quality can be affected by the conditions in which it is gathered: available light sources and type of imaging equipment used to collect the image. It can also be affected by the method in which an image is stored: often an image must be compressed due to memory limitations. Noise that is seen in digital images is usually the result of non-uniformity of the image due to the imaging process. Image enhancement techniques are used to enhance the signal to noise ratio and better define certain features of an image by modifying its intensities and colors. User-dependent (20) and user-oriented (13; 14) approaches to image enhancement have already been introduced in literature. However, an attempt to use reinforcement learning has not yet been made.

We have limited the scope of this paper to image contrast/brightness. Contrast refers to the relative variation of intensity in an image. It is typically measured as the ratio of the change in intensity to the average intensity. Simple intensity transformations known as ”point operations” can achieve dramatic contrast effects. It is these enhancement operators, on which we focus.

4. Reinforced Image Enhancement

Figure 2 shows the steps of the learning algorithm. The steps are summarized below.

4.1. *Selection of an Original Image*

An image is randomly selected from a set of twelve gray-scale standard test images. These images do not contain any *visual arguments* - information to be interpreted by the expert. This study is only interested in the subjective evaluation of an image, not the processing of information delivered by that image. The original image is not shown to the observer, so that the evaluation of the enhanced image is independent of the quality of the original image.

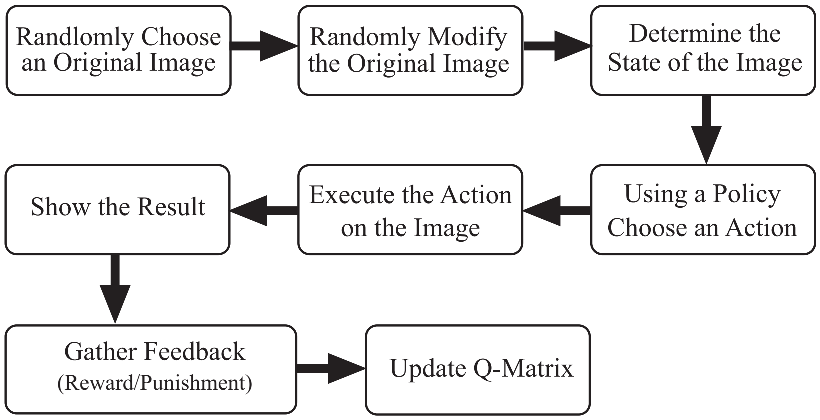

Fig. 2. Algorithm overview

4.2. *Modification of the Original Image*

The original image is modified to vary its quality. It would be ideal to have a suite of images that cover the entire gamut of contrast values. Instead of using the same set of static images for testing, the agent learns on images that are not necessarily the same for each test. For this reason, we have chosen to modify the original images at run time. The modification is a simple a linear point transformation. This function alters the image in a harsh way to produce a wider range of images. At the time of image modification, the program chooses 2 random values, which are determined to be the new low and high values of the image histogram. Histogram values are then scaled linearly between these values, often resulting in images with poor contrast, or images that are too light or too dark.

4.3. *Determining the Image State*

It is necessary for the Q-learning algorithm to have a numerical measure of the state of both the bad image and the good image. The state should best represent the contrast and brightness of the image. The state was thus determined using the histogram of the image. Four key measures of the image state were selected. These were the brightest (g_{max}) and darkest (g_{min}) value in the image and the position of the peak ($g_{h_{max}}$) of the histogram. The remaining decision to make was the precision of each of these measurements.

A measurement precision covering the entire range of [0,255] for gray-scale images would give unambiguous results, but this would produce too many states. To see results within 100 iterations, the number of states had to be reduced significantly. The final prototype simply used 3 values for each of the measures: *low*, *medium* and *high*. The impossible states were eliminated, and the possible permutations of the 3 measurements resulted in 10 possible states. The states were organized such that low states represented the darkest images and high states represented bright images. A middle range state represented a mid-range image, but did not necessarily imply the level of contrast. An example of the calculation of the state from the image histogram is given in Figure 3.

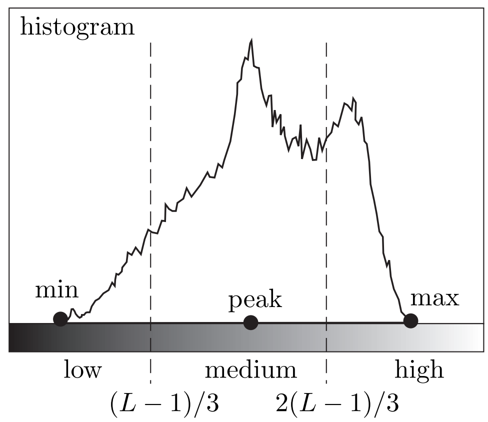

Fig. 3. Determination of states from histogram

4.4. *Policy Selection*

A policy is the method the agent takes to select an action to execute in order to change the current state. The agent must make a trade-off between immediate return and long-term value. The agent that did not explore unseen states and only attempted to maximize its return by choosing what it already knew would be a poor learner (18). There needs to be a balance between exploration of unseen states and exploitation of familiar states.

Initially, ε-Greedy policy selection was implemented, as it is one of the most commonly used method in reinforcement learning applications (18). A greedy action is an action whose estimated value is the greatest. If the greedy action is chosen, the agent is said to be exploiting. If, instead, a non-greedy action is selected, the agent is said to be exploring. The ε-Greedy method selects the action with greatest value most of the time, with probability $(1-\varepsilon)$. With the small probability ε , it selects a random action. Thus the method satisfies the balance of exploration and exploitation.

One major drawback of ε-Greedy action selection is that when the agent explores, it chooses equally among non-ideal actions. This is undesirable when certain actions are much lower valued than others. A policy called ”softmax action selection” solves this problem by varying the action probabilities as a graded function of estimated value. The greedy action is given the highest selection probability, but the other actions are ranked and weighted according to their value estimates. The most common softmax method uses a Gibbs or Boltzmann distribution (18). The number of state changes accumulated must be known to the agent. The agent will then choose action a on iteration t with probability

$$P(a|t) = \frac{e^{\frac{Q_{t-1}(a)}{\tau}}}{\sum_b e^{\frac{Q_{t-1}(b)}{\tau}}}, \tag{1}$$

where τ is a positive parameter called the *computational temperature*. Higher temperatures will result in all actions being equally probable. As τ approaches 0, Softmax error selection becomes the same as greedy action selection.

The largest challenge found in implementing this type of policy selection was finding an optimal value of temperature. This was dependent on the magnitude of Q-values. A temperature value of $\tau = 0.1$ was found to make the action selection very exploratory. Higher temperatures, $\tau > 0.1$ also resulted in exploratory performance, but the probabilities between actions were so similar that there was little difference noticed in temperature values above $\tau = 0.1$. This was due to the small magnitude of the Q-values, which resulted from the arbitrary choice of reward values and the value of the learning rate parameter. Raising the learning rate parameter had the effect of increasing the magnitude of the Q-values. This also changed the performance of Softmax action selection with respect to temperature sensitivity. A value of $\tau = 0.01$ resulted in the Softmax policy having action selection comparable to the $\varepsilon = 0.1$ ε-Greedy method, but with weighted non-ideal actions. More exploratory performance on early iterations and more greedy performance on later iterations were desirable. The most successful results, from what was observed during development, was the use of a temperature function that decreased rapidly for the first group of iterations and then levelled out at a point after approximately a quarter of the maximum number of iterations. Such a relationship can be described by the function:

$$\tau = \frac{\tau_0}{\sqrt{\kappa}} \tag{2}$$

where κ is the iteration, and τ_0 is an exploratory (high) starting temperature.

Figure 4 shows the plot of such a function with $\tau_0 = 1$ over 50 iterations. The agent is expected to perform very exploratory over approximately the first 10 iterations, but behave greedily over the remaining iterations.

4.5. *Taking Actions on the Bad Image*

We selected a set of point operation functions based on their visible effect on the image. One action should be selected to correct for any of the following four basic problems:

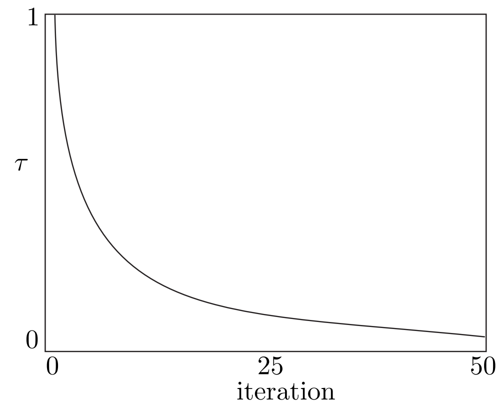

Fig. 4. Boltzmann temperature function

(1) Image is too bright
(2) Image is too dark
(3) Image is too washed out or gray (poor contrast)
(4) Image has too much contrast

Each of these problems can be corrected using a unique image transformation. For this reason, four actions were selected, one of which would be able to correct for any of the four possible situations above. The final selection of the action set is shown in Figure 5 (g: current gray levels, g': modified gray levels).

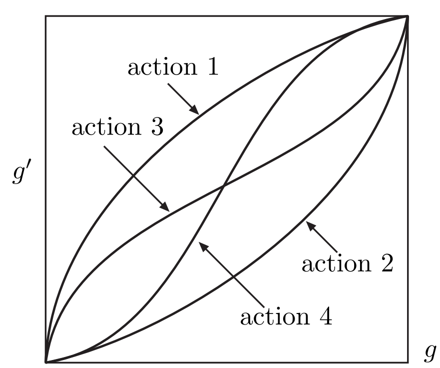

Fig. 5. Action set

The parameters of each action were selected so that the user could readily see the effect of the operation. These values were primarily selected by trial and error. The observers evaluated the impact of the transformation visually, and selected appropriate values subjectively. Although this seems to eliminate some of the objectiveness of the agent, constant transformations must be selected to give the agent some idea of where to start.

4.6. *Reward and Punishment*

A standardized method of rewards/punishments needed to come from user interaction. The Mean Opinion Score (MOS) was selected for measuring the user's evaluation of each modified image. MOS allows the user to express their opinion on a 1-5 scale, 1 being the least desirable and 5 being the most desirable quality. Table 1 demonstrates the implementation of MOS in the agent. Magnitudes were kept between 0 and 1 for simplicity.

MOS Score	Description	Reward/Punishment
1	Much Worse	-0.4
2	Slightly Worse	-0.2
3	Same	0
4	Slightly Better	0.2
5	Much Better	0.4

Table 1. MOS implementation in the reinforcement learning agent

4.7. *Update the Q-Matrix*

After each action and subsequent interaction with user, the agent uses its reward and the current values of the matrix related to both old and new states to update the Q-value matrix. The notation $Q(a, i)$ is used to denote an action-value or Q-value for a doing a particular action, a, from state i. Q-values are updated after every transition from state i to state j when action a is performed. The update equation is:

$$Q(a, i) \leftarrow Q(a, i) + \alpha \left(R(i) + \max_{a'} Q(a', j) - Q(a, i) \right), \tag{3}$$

where a' is the action taken from state j. For both development and demonstration purposes, the Q-matrix was represented as a gray-scale image.

5. Implementation and Results

5.1. *User and Development Interfaces*

The user interface acts as the intermediary between the users and the content of the application. The users must interact with the interface to view the images, assess the image quality and provide rewards or punishment for the agent. It is important for this process to be simple and intuitive in order to make the experience easy for the user. The more difficult the interface is to use, the more likely the user will be to make mistakes and, therefore, provide incorrect information to the algorithm. Any incorrect or biased selections will interfere with the algorithm and cause false results.

A development interface was also created solely for development purposes (Figure 6). This interface would not be used for testing, as if the results were shown to the test participant throughout the test, this could affect the user's subjectivity.

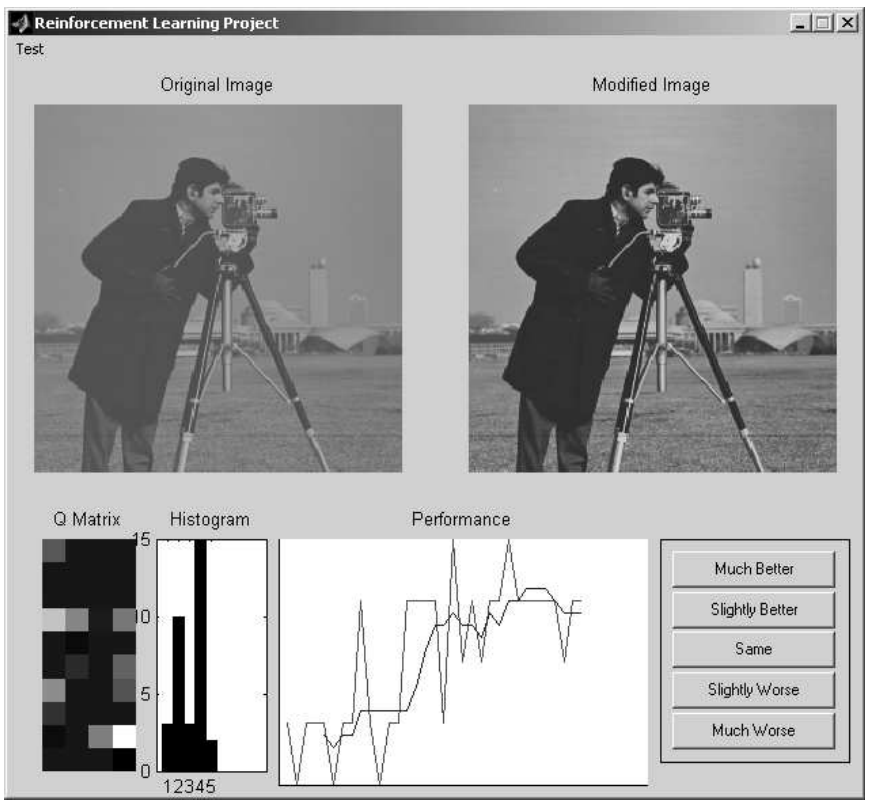

Fig. 6. Development interface

5.2. *Measuring Success and Performance Verification*

The foremost research objective is to determine if reinforcement learning is an appropriate technique for the automation of the subjective image enhancement. In order to come to a conclusion, evaluation metrics are required.

Four deterministic metrics were established: rewards to punishments ratio

(R/P), running average (RA), total average (TA), and the histogram. These are readily computable and taken together, provide a way to determine if the agent has the potential to solve the subjective image enhancement problem.

- **Histogram** The histogram is simply a graphical representation of the frequency of each reward or punishment value. More rewards than punishments is desirable, however, the histogram is not an ideal metric because it excludes the iteration at which the value was given. For this reason, the histogram is a useful overview of the results, but is not considered conclusive in measuring algorithm success. An example histogram is shown in Figure 7.

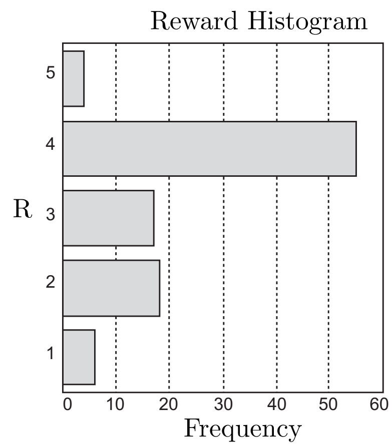

Fig. 7. Histogram example

- **Reward to Punishment Ratio (R/P)** The reward to punishment ratio is defined as:

$$\frac{R}{P} = \frac{\text{Number of rewards up to current iteration}}{\text{Number of punishments up to current iteration}} . \quad (4)$$

  Rewards are expected to increase with the number of iterations, and a ratio R/P greater than 1 indicates that the agent is being rewarded more than punished. This ratio is also used to indicate the point in testing at which R/P stops increasing. This indicates the maximum number of iterations required for the agent to reach is maximum learning potential. An example of a successful R/P plot from our test series is shown in Figure 8.
- **Running Average** The running average is calculated over the past ten iterations of the algorithm. It is defined as:

$$RA = \frac{\sum_{N-10}^{N} r_n}{10} , \quad (5)$$

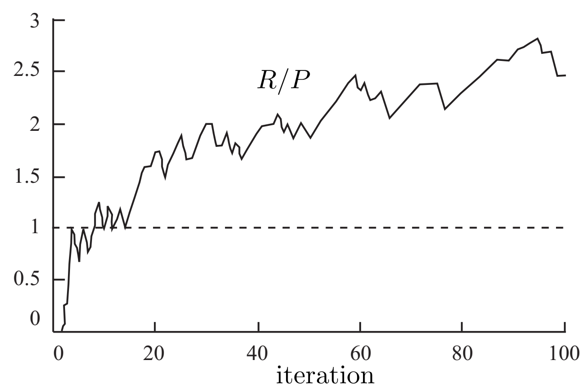

Fig. 8. Example R/P plot

where N is the current number of iterations and r_n is the reward or punishment value (user input) received at iteration n.
An RA with a value above 3 indicates that the agent has been rewarded more than punished over the past ten iterations. Ideally, the RA increases steadily as the iterations increase, up to the maximum iterations required by the agent in order to learn the optimal image.

- **Total Average** The total average is calculated over all of the past iterations. It is defined as:

$$TA = \frac{\sum_N r_n}{N}, \tag{6}$$

where N is the current number of iterations and r_n is the reward or punishment value (user input) received at iteration n.
Similar to the RA, a TA with a value above 3 indicates more rewards than punishments over all of the iterations. Ideally this metric increases over all of the iterations.

5.3. *Test Series*

In order to apply the discussed metrics, data is required from users. This data was obtained during two computer testing sessions. Each session had four observers test the algorithm via the user interface.

The goal of the first testing session was to measure the success of the reinforcement learning algorithm using the ε-Greedy action selection policy. Each test in the first session was composed of 100 iterations. The Softmax (Boltzmann) action selection policy was implemented after the first testing session and required testing in order to compare it with the ε-Greedy policy. We initiated a second testing session to compare the results using both action selection policies, as well as measure the results with a fewer number of iterations. Each test in the second session was composed of 50 iterations. In the second session each observer completed the test using the ε-Greedy policy first, then repeated the test using the Boltzmann policy. The algorithm was evaluated in terms of the metrics discussed in Section 5.2.

5.3.1. *Algorithm Metric Results*

The summarized results of the first testing session are shown in Table 2. The results are positive in all four cases. The mean R/P ratio is close to 2, when a result of 1 would indicate equal rewards and punishments. This value indicates that the agent is successful in enhancing images to the liking of the individual users. The results for users 1 and 3 are significantly better than those of 2 and 4. When we examine the histogram and plots of R/P, RA and TA, we see that the ratio of rewards to punishments is above one after fifteen to twenty iterations, the running average is at or above 3 the majority of the time, the total average is at or above 3 after approximately 20 iterations and each histogram shows more rewards than punishments.

Table 2. Results for first testing session. U_i represents the score from user i. U_{mean} represents the mean score from all users.

Metric	U_1	U_2	U_3	U_4	U_{mean}
R/P Ratio	2.46	1.05	2.45	1.39	1.84
Running Average (final 10)	3.20	3.30	3.70	3.00	3.30
Total Average	3.34	3.03	3.35	3.11	3.21

The results from the second testing session are still positive, but less decisive. Here, the number of iterations has been reduced to 50 from 100. The ε-Greedy policy results are shown in Table 3. For the ε-Greedy policy tests, users 2 and 4 show positive results. This has an effect on the mean R/P ratio, which is significantly greater than 1. When we examine the histogram and plots of R/P, RA and TA, we see that for users 2 and 4, their histograms show more rewards than punishments, the reward to punishment ratio is increasing and above 1 the majority of the time and both the total and running averages are at or above three most of the time. User 1's results are less positive, as the reward to punishment ratio is below 1 most of the time. However, the reward to punishment ratio and the total average are generally increasing over the iterations. This indicates that even when the agent is running with a R/P ratio less than 1, learning is still exhibited. User 3's results are the least positive: while the total and running averages are generally at or above 3, the reward to punishment ratio is always below 1. Overall, we see that the mean total average score is 3.78, well above the middle MOS score of 3.

The results for the Softmax (Boltzmann) policy are summarized in Table 4. Once again, the number of iterations has been limited to 50. The mean R/P ratio

Table 3. Results for testing session, ε-Greedy policy. U_i represents the score from user i. U_{mean} represents the mean score from all users.

Metric	U_1	U_2	U_3	U_4	U_{mean}
R/P Ratio	0.79	2.11	0.81	1.53	1.31
Running Average (final 10)	2.40	3.30	3.10	3.80	3.15
Total Average	2.88	3.30	2.90	3.20	3.07

Table 4. Results for testing session, Softmax (Boltzmann) policy. U_i represents the score from user i. U_{mean} represents the mean score from all users.

Metric	U_1	U_2	U_3	U_4	U_{mean}
R/P Ratio	1.32	1.78	0.29	0.50	0.97
Running Average (final 10)	3.10	2.50	2.40	2.80	2.70
Total Average	3.20	3.18	2.56	2.76	2.92

is 0.97, nearly 1. This indicates that on average, an almost equal amount of rewards and punishments were given. The mean total average score is 2.92, slightly below the MOS score of 3. This result is not as strong as that obtained using the ε-Greedy policy, but remains nearly average. Two of the users have exhibited strong results, while the other two exhibit poor results. Examining the histogram and plots of R/P, RA and TA, users 1 and 2 have positive results with the reward to punishment ratio above 1 and increasing after 20 iterations. The total and running averages are at or above 3 the majority of the time. User 4's results show an increasing reward to punishment ratio, as well as increasing running and total averages. However, the reward to punishment ratio is always well below 1 and the total and running averages are below 3 most of the time. This is similar to a result obtained using the ε-Greedy policy where the agent is punished more than rewarded, yet shows an increasing R/P ratio and is thus learning. User 3 shows negative results. The reward to punishment ratio is below 1 all the time and decreasing. The running and total averages are below 3 and decreasing. In general the negative results are stronger for the Softmax (Boltzmann) policy than the ε-Greedy policy.

6. Conclusions

The foremost conclusion from testing the algorithm design is that reinforcement learning has the potential to solve the problem of subjective image enhancement. The conclusion comes from the following observations:

- Of the eight users tested, the majority had positive results based on the algorithm metrics.
- The users with less positive results generally indicated some degree of algorithm

success.

- These positive results were arrived at with 100 or fewer iterations, meaning the learning process can take place in a reasonable amount of time.

However, the results are unable to conclude if the Boltzmann or the ε-Greedy policy is preferable. Further testing is required with a larger sample space in order to make this decision. The literature does not provide any proof of the superiority of one policy over another. The performance of the policies may depend on human factors and the task at hand(18). Both policies should be explored further, with additional testing and with more advanced reinforcement learning methods. These methods must be investigated before arriving at a definitive conclusion about the precise elements of reinforcement learning that should be applied to subjective image enhancement.

7. Future Considerations

The scope of this project only included the use of Q-learning as a reinforcement learning algorithm. There are other such algorithms more widely used in this field, such as SARSA and $TD(\lambda)$, which should be tested and compared with the results for Q-learning.

So far, we have limited the agent's image enhancement ability to modifying the contrast/brightness of an image. In order to determine a truly optimal image, the algorithm should account for image characteristics such as smoothness and sharpness. This will be subject to future investigations.

References

1. Algazi R. et al., "Quality Measurement and Use of Pre-Processing in Image Compression". Signal Processing 70, pp. 215-229, 1998
2. Avcibas, I , Sankur, B., Sayood, K., "Statistical Evaluation of Image Quality Measures". Submitted to Journal of Electronic Imaging, 2001
3. Betts, B.J. et al., "Image Quality in Digital Mamography". Stanford University, USA, Report to the Army Medical Research and Material Command, Compression and Classification of Digital Mammograms for Storage, Transmission, and Computer Aided Screening, 1998
4. Bhanu, B. et al., "Adaptive target recognition". Machine Vision and Applications 11, pp. 289-299, 2000
5. Carrino, J.A., "Digital Image Quality: A Clinical Perspective". In: Quality Assurance: Meeting the Challenge in the Digital Medical Enterprise, Bruce Reiner, Eliot Siegel and John Carrino (Eds.), SCAR University Primer 3, ISBN: 0-9706693-2-1, 2002
6. Contin L. et al., "Subjective Evaluation of MPEG-4 Video Codec Proposals: Methodological Approach and Test Procedures". Signal Processing: Image Communication 9, pp. 305-325, 1997

7. Gijbels F. et al., "The Subjective Image Quality of Direct Digital and Conventional Panoramic Radiography". Clin Oral Invets 4, pp. 162-167, 2000
8. Janssen, T.J.W.M., Blommaert, F.J.J., "A Computational Approach to Image Quality". Displays 21, pp. 129-142, 2000
9. Jurik, A.G. et al., "Image Quality and Dose in Computed Tomography". European Radiology 7, pp. 77-81, 1997
10. Laskey, W.K. et al., "Image Quality Assessment: A Timely Look". Catheterization and Cardiovascular 46, pp. 125–126, 1999
11. Malo, J. et al., "Subjective Image Fidelity Metric Based on Bit Allocation of the Human Visual System in the DCT Domain". Image and Vision Computing 15, pp. 535-548, 1997
12. Martin, C.J., Sharp, P.F., Sutton, D.G., "Measurement of Image Quality in Diagnostic Radiology". Applied Radiation and Isotopes 50, pp. 21-38, 1999
13. Munteanu, C., Rosa, A., "Evolutionary Image Enhancement with User Behavior Modeling". Proceedings of the 2001 ACM symposium on Applied computing, ACM Press, pp. 316 - 320, 2001
14. Picard R.W. et al., "Modeling User Subjectivity In Image Libraries". IEEE Int. Conf. On Image Processing, pp. 777-780, 1996
15. Ribeiro, C., "Reinforcement Learning Agents". Artificial Intelligence Review 17, pp. 223-250, 2002
16. Roman-Roldan R. et al., "A Measure of Quality for Evaluating Methods of Segmentation and Edge Detection". Pattern Recognition 34, pp. 969-980, 2001
17. Scheifele, C., "Subjective Image Quality in Digital Panoramic Radiography in Relation to Patient Exposure". International Congress Series 1230, pp. 1291-192, 2001
18. Sutton, R.S., Barto, A.G., "Reinforcement Learning: An Introduction". MIT Press, Cambridge, MA, 1998
19. Tizhoosh, H.R., Michaelis, B., "Subjectivity, Psychology and Fuzzy Techniques: A New Approach to Image Enhancement". NAFIPS'99, New York, USA
20. Tizhoosh, H.R., "Observer-Dependent Image Enhancement". In: Fuzzy Filters for Image Processing, M. Nachtegael, D. Van der Weken, D. Van De Ville & E.E. Kerre (Eds.), Springer, Studies in Fuzziness and Soft Computing, 2002
21. Van Dijk, A.M., Martens, J.-B., "Subjective Quality Assessment of Compressed Images". Signal Processing 58, pp. 235–252, 1997
22. Vogiatzis, D., Stafylopatis, A., "Reinforcement learning for rule extraction from a labeled dataset". Cognitive Systems Research 3, pp. 237-253, 2002
23. Watkins, C.J.C.H., "Learning from Delayed Rewards". PhD thesis, Cambridge University, Cambridge, England, 1989