

# Identifying Features of Video Games That Are Most Suited to Machine Learning

Peter Ting

Graduate, School of Mathematics and Computer Science, University of Wolverhampton, Wolverhampton,  
United Kingdom

**ABSTRACT:** Deep Learning neural networks have been trained to play a wide variety of video games, including Ms. Pac-Man, and are widely used in deep reinforcement learning research. In addition, machine learning AI has been applied to many different video games, but with differing effectiveness. Video games are often used as a testbed for systems that need to be trained in a simulated environment before real-world usage, therefore making efficient training important. In this paper, we investigate and discuss the features of video games that affect the effectiveness of machine learning algorithms used to play them.

**KEYWORDS:** Machine learning, Deep learning, Convolutional neural network, Video games.

## I. INTRODUCTION

Deep learning neural networks are traditionally very powerful at recognising patterns and solving complex tasks that previous AI systems could not. Their use in playing video games extends that to a multifaceted scenario and can be used as a basis to test a method's effectiveness with complex situations[1]. Through prior research in board and video games, such as Chess, Ms. Pac-Man, and Go, these neural networks have proven they are able to handle many different tasks[2].

An example of video games being used to train AI systems is the use of TORCS (The Open RacingCar Simulator) used to train self-driving cars before being used to drive on real roads [3]. By using a video game simulator, the network can be trained in safer conditions and without having to reconfigure physical hardware to test different approaches. A similar usage is the use of simulators built on top of game engines, such as Unreal Engine 4, which can be used to enhance computer vision capabilities, such as unmanned drone flight[4].

Results that exceed traditional methods are generally tailored to their solution such as the AI AlphaGo with both Go [5]andOthello [6]. More recently, however, networks have been trained on these games with no human-engineered features, and have exceeded previous solutionthrough more advanced networks and self-training, such as AlphaGo Zero, the successor to the previous AlphaGo Master [7]. Even with these advances, general deep learning algorithms may have different levels of efficacy when presented with different scenarios with different features. In this paper, we will evaluate this behaviour using a variety of Atari video games.

To do this, we first evaluated key features of each game, based on their control scheme, information available to the player, reward strategy, and complexity for a human. A neural network was set to play each game for a fixed number of iterations with final scores of each game recorded. These scores were then evaluated to see what features appeared to have a positive or negative effect on the score of the AI, and therefore their use in training the network.

The comparisons were not drawn from how well the neural network played an individual game, but how well it did relative to itself playing other video games, and the amount that the network was able to improve its score.

## II. RELATED WORK

The use of neural networks to play video games, specifically Ms. Pac-Man, has been written about from at least 2001 [8]. For an example of the level of intelligence of an AI that can be feasibly trained, one paper utilising a neural network was able to train an AI to the level of a human novice, capable of clearing the first level, but also making frequent mistakes [9]. A more successful example was shown by the team from "Maluuba", who were able to produce an AI system that was able to achieve a score of "999,990" and beyond in Ms. Pac-Man [10].

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

Game-playing AI has always been an important part of AI development, leading to large amounts of innovation in AI research [11]. Development in machine learning is no exception, a recent example in video games being the Dota 2 bot competition with the AI winning games against human players [12]. These human-versus-AI events are helpful in the benchmarking of AI progress and complexity, with additional speculation as to whether AI's can exceed human knowledge [13].

Certain video games also pose a challenge to traditional machine learning methods. A prime example would be "Montezuma's Revenge", a complex adventure style game. The neural network developed to beat the game had to be given specific domain knowledge in order to be able to beat it, as a traditional setup produced no real results [14]. The main problem cited was the lack of feedback, due to very few rewards from simply using random inputs.

This lack of feedback is seen in other research, where video games that have immediate feedback are easier to train than those with a delay [15]. Some techniques were found to mitigate these issues, such as the use of Monte-Carlo Tree Search Planning which has been tested on the same Atari video games used in these experiments [16]. These methods have their own downsides, however, such as needing more information about the game than a human player, which could restrict its usage, therefore more research into the features of the video games that affect more traditional training is needed to fully investigate this.

### III. EXPERIMENTAL SETUP

In order to test a variety of video games, OpenAI's library of Atari video games was used, due to standardised inputs and display. A deep convolutional network was set up to play each game, using the controls and game screen as inputs and the game score as a reward. Image pre-processing was used to minimise extraneous data for the algorithm by first reducing the size and then restricting the colours to greyscale. Each experiment was run for 2'000'000 iterations, except for Ms. Pac-Man, which was used to gauge an appropriate number of iterations.

### IV. EXPERIMENTAL RESULTS

#### A. Ms. Pac-Man

An initial experiment was performed with Ms. Pac-Man to establish a baseline, to see what issues would arise when training, and what a useful number of iterations would be to see the training progress of the AI.

The game was first trained for 2'000'000 iterations to see the initial effects of training on the network and then trained to a total of 4'000'000 iterations. Results of each game were plotted on a graph, with a moving average line to see the overall trend for this and further experiments.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

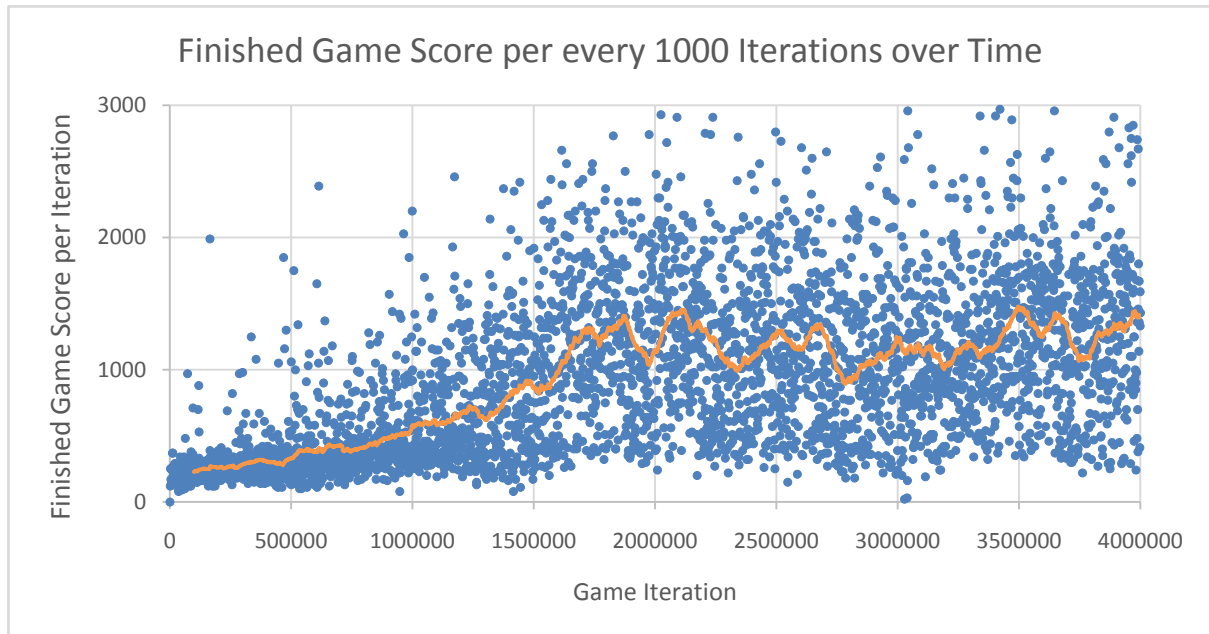


Figure 1. Graph of Ms. Pac-Man Experiment

Table 1. Table of Ms. Pac-Man Experiment Results

Ms. Pac-Man	Score
Mean Score (Last 50 Iterations)	1217
Mean Score (Total)	947
Standard Deviation (Last 50 Iterations)	709.84
Standard Dev Total (Total)	640.50
Max Score	5210

A general trend upwards can be seen from Figure 1, with a plateau around 2'000'000 iterations. From Table 1, the average score at the end of training using the last 50 values was 1221, and the total average was 947. which was accumulated from traversing around half the playable area. This was a relatively low score, on par with a first-time player of the game.

## B. Breakout

Breakout differs from Ms. Pac-Man in that the player's input is not as tightly coupled to the reward, due to the travel time of the ball bouncing between the paddle and the bricks. The game also requires knowledge of the current speed and direction of the ball, which is a unique feature of Breakout.

## International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

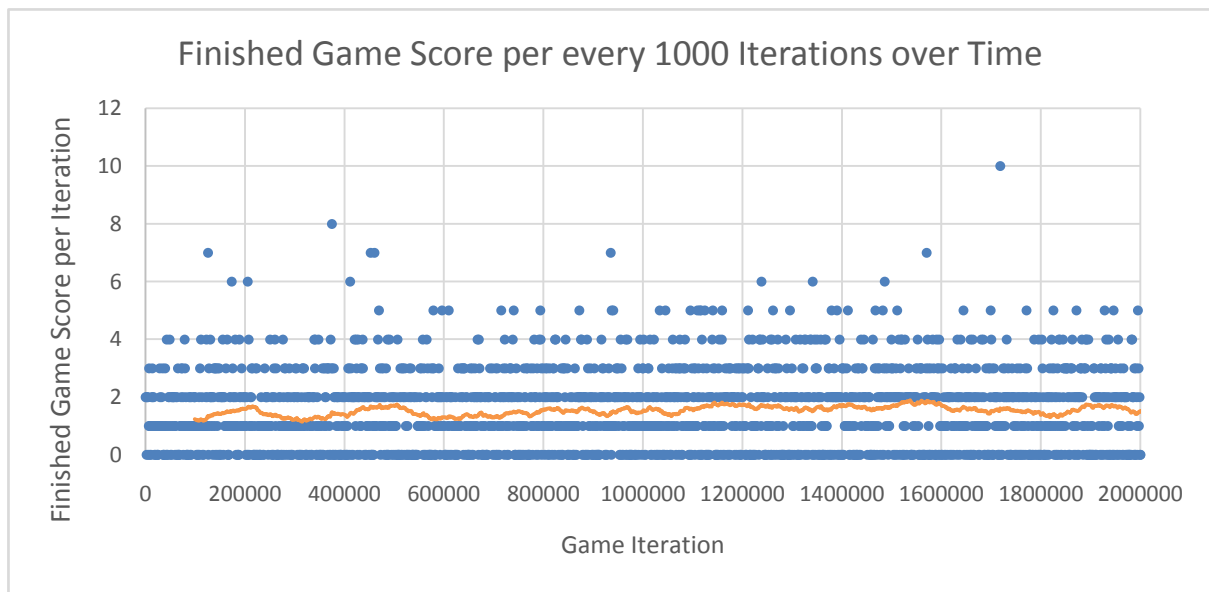


Figure 2. Graph of Breakout Experiment

Table 2. Table of Breakout Experiment Results

Breakout	Score
Mean Score (Last 50 Iterations)	2
Mean Score (Total)	2
Standard Deviation (Last 50 Iterations)	1.52
Standard Dev Total (Total)	1.36
Max Score	10

**Ошибка! Источник ссылки не найден.** shows a mostly flat learning rate, which is also shown in **Ошибка! Источник ссылки не найден.**, indicating very little change due to training. Overall, this experiment was not successful in the playing Breakout and observed AI behaviour showed little difference between a trained network and random inputs.

### C. Pong

Pong has similar features to Breakout but is also the only competitive game where the agent competes against the inbuilt AI of the game. This game also requires knowledge of the speed and direction of the ball, but to score points, knowledge of the opponent's position is required as well.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

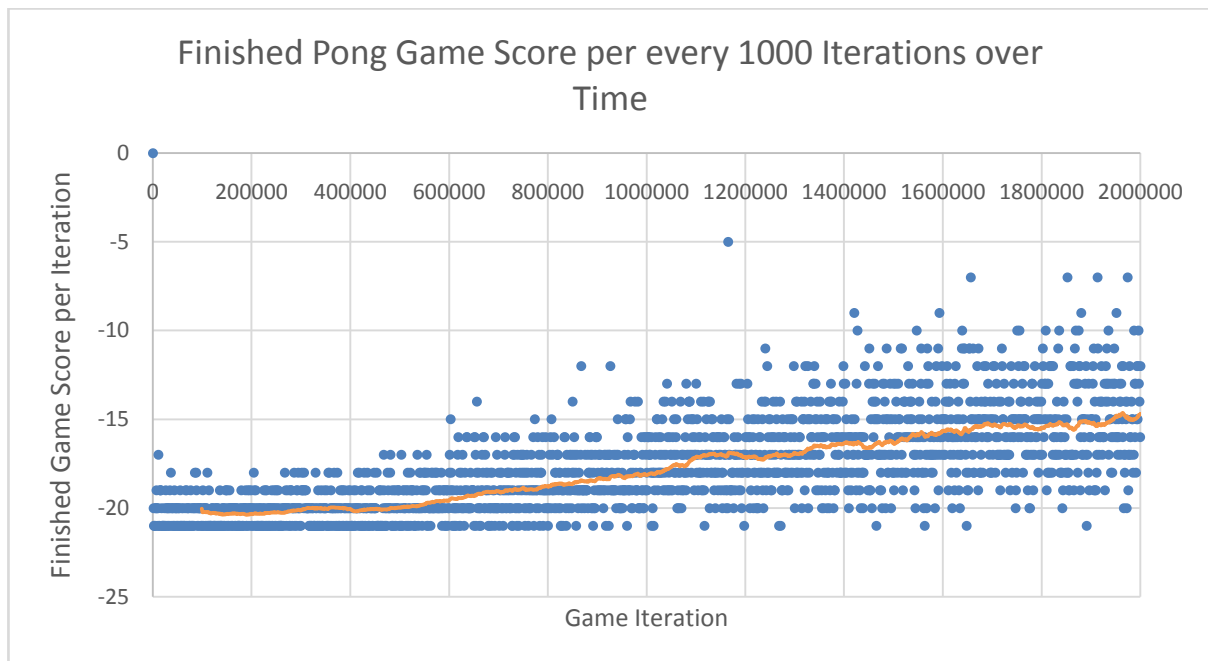


Figure 3. Graph of Pong Experiment

Table 3. Table of Pong Experiment Results

Pong	Score
Mean Score (Last 50 Iterations)	-13
Mean Score (Total)	-16
Standard Deviation (Last 50 Iterations)	2.88
Standard Dev Total (Total)	3.44
Max Score	-2

Unlike Breakout, score generally trended upwards as seen in Figure 3. Due to the nature of Pong, the scores are presented with the agent's score minus the game AI's score, and so a loss is shown with a negative number. The agent managed to increase its score against the opponent, but never managed to win a game.

#### D. Qbert

Qbert is a platformer style game with the objective of jumping on all the squares of the game board arranged in a pyramid fashion. The gameplay style is most like Ms. Pac-Man, with points given for moving to new squares, and with hazards to be avoided. This game adds additional dangers with random inputs, such as being able to fail by jumping off the play area.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

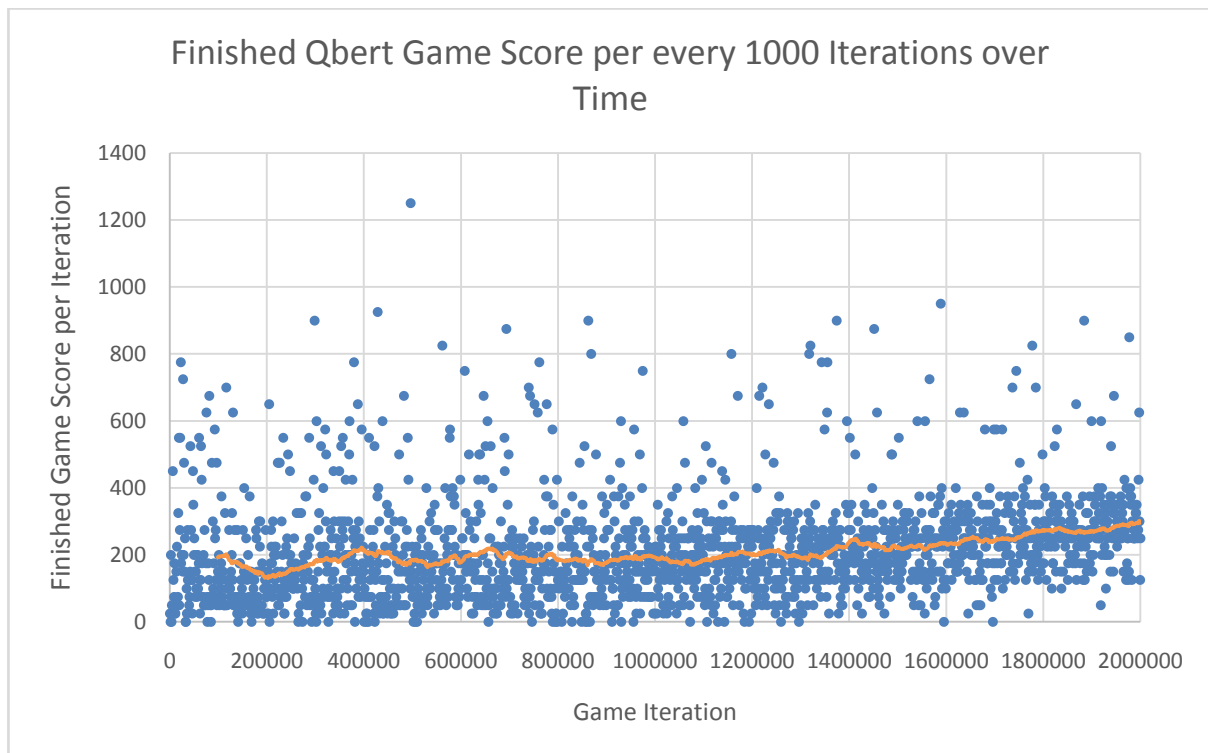


Figure 4. Graph of Qbert Experiment

Table 4. Table of Qbert Experiment Results

Qbert	Score
Mean Score (Last 50 Iterations)	269
Mean Score (Total)	257
Standard Deviation (Last 50 Iterations)	74.30
Standard Dev Total (Total)	116.87
Max Score	1250

This graph (Figure 4) shows a slight upward trend, suggesting that the agent managed to improve its score through training, but not to a lesser degree than previous games. It is likely the anomalous max score (Table 4) is due to random bonus items on the map, rather than the agent playing the game well.

### E. Space Invaders

Space Invaders requires the agent to aim and fire at descending enemies, with points scored for each successful hit. Due to the player being able to control their position, enemies changing position, and the travel time of the bullet, this game is significantly more complex than Ms. Pac-Man or Qbert.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

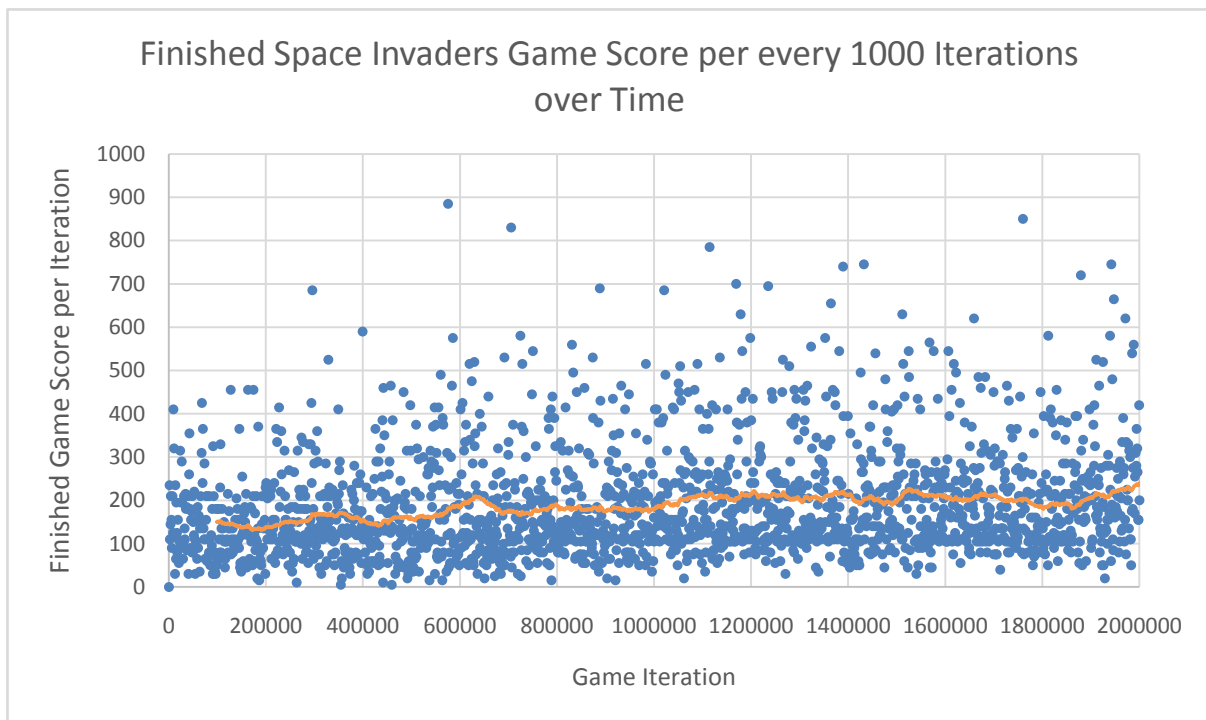


Figure 5. Graph of Space Invaders Experiment

Table 5. Table of Space InvadersExperiment Results

Space Invaders	Score
Mean Score (Last 50 Iterations)	208
Mean Score (Total)	190
Standard Deviation (Last 50 Iterations)	109.27
Standard Dev Total (Total)	122.16
Max Score	885

Similar to Qbert, a slight upward trend in score was seen (Figure 5)for Space Invaders. Due to the delay between the bullet being fired and it hitting an enemy, which is the only action which grants rewards, it’s likely the agent had difficulty in assigning useful inputs to rewards.

### F. Asteroids

Asteroids required the agent to fire at incoming asteroids, and also pilot a ship with rotation and thrusting inputs applied to move the player ship. This was the most complex game available for the agent, as it required knowledge of direction and speed of all in-game objects, complex movement controls, and many chances for failure.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

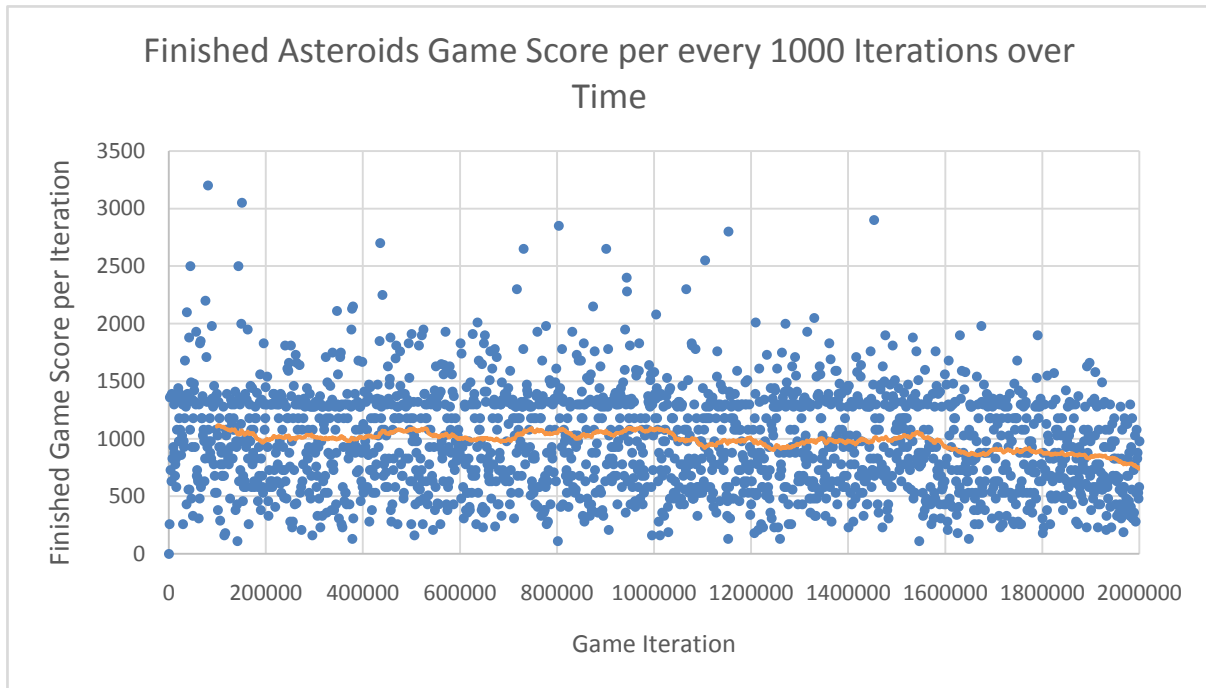


Figure 6. Graph of Asteroids Experiment

Table 6. Table of Asteroids Experiment Results

Asteroids	Score
Mean Score (Last 50 Iterations)	769
Mean Score (Total)	865
Standard Deviation (Last 50 Iterations)	329.16
Standard Dev Total (Total)	404.25
Max Score	3200

Asteroids was one of only two experiments to achieve a negative trend in score (Figure 6). Due to the unique properties of Asteroids, such as the increased chance of failure, complex movement controls, and multiplying obstacles, it is likely that Asteroids presented a difficult scenario for the network to train against.

### G. Pinball

Pinball is a simulation of a standard pinball machine on the Atari, with the only inputs available being pushing the left or right flipper. This gives the agent little control over the state of the game for most of the playtime.



# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

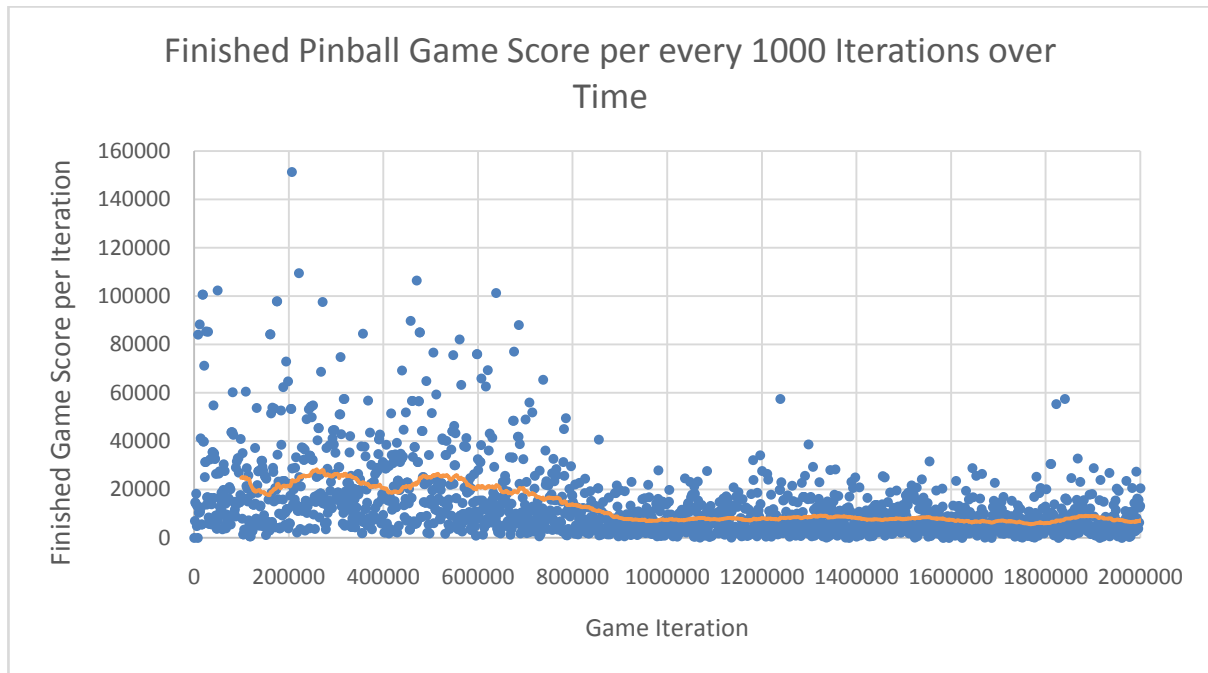


Figure 7. Graph of Pinball Experiment

Table 7. Table of Pinball Experiment Results

Pinball	Score
Mean Score (Last 50 Iterations)	6622
Mean Score (Total)	12713
Standard Deviation (Last 50 Iterations)	4606.16
Standard Dev Total (Total)	14653.50
Max Score	151278

Pinball, like Asteroids, saw a downward trend in score over time. Due to the agent’s actions having very little effect on the game state, it was likely the rewards were mostly confusing for the network, making the game difficult to train against.

## V. DISCUSSION

In order of score improvement, the games were Ms. Pac-Man Pong, Qbert, Space Invaders, Asteroids, and finally Pinball.

Starting with Ms. Pac-Man, the main features of the game for the network was the number of rewards available, and the close correlation of inputs to rewards. Pressing a direction is more likely to move Ms. Pac-Man into a pellet than not, and therefore more likely to lead to better rewards with more inputs. Main limiting factors in this would be the ghost enemies which cause the game to end, and the difficulty in avoiding them

In Qbert, the player must choose where to jump. The movement key corresponds to a block to jump to, which gives an immediate score reward. Negative features that skewed the training may have been the large bonuses available that appear randomly, which skewed the network towards those inputs in future attempts, even when it would be detrimental. The presence of enemies is also likely to contribute to the overall low score trend.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

The main gameplay in Space Invaders is avoiding and firing at enemies, where pressing the fire button will fire a travelling bullet towards the enemies. Due to the bullet's travel time and the ability to miss, this means that the actions the AI takes do not immediately correlate to an appropriate reward, which likely had a negative effect on the score. However, the game also featured many opportunities to obtain rewards, which may have been enough to allow the score to trend upwards. However, the presence of enemies and a hard time limit were likely factors in keeping the trend upwards minimal.

With the next two video games, training the network was actively detrimental to the overall scores.

Starting with Asteroids, a general downward trend of the score is likely due to the controls and nature of the game. Firstly, a bullet being fired may not hit an asteroid for some time, meaning a delayed reward. Secondly, staying alive in the game is a much more difficult task, due to the momentum of the ship and the constantly moving hazards. Additionally, when a bullet hits an asteroid, it splits, creating more obstacles that move faster, causing an increase in difficulty over time. While it is intuitive for a human to rotate the ship and thrust in the opposite direction to slow the ship, this is likely difficult for a network to learn given random inputs.

The least effective game was Pinball, likely due to several factors, including the randomness and lack of control inherent to the game. Interestingly the network developed certain quirks in behaviour while training on this game, such as when the ball hit a pin in the machine that made the screen flash white, the network would then start hitting the flippers even if the ball was nowhere near them. Due to the rewards being gathered by the ball hitting objects in the machine, the flash was likely strongly correlated with a reward, and so any inputs used at the time were taken as "correct" despite it not contributing.

The most interesting game was Pong which has similar gameplay to Breakout but had an upwards trend of score. Since the act of scoring a point is not directly tied to any particular action of the paddle, this appears to lack the features required for easy training.

One probable reason for the network being able to learn is that, besides the ball, there is also the opposing AI's paddle. This paddle's movement matches the ball almost exactly unless the ball is moving too fast. This means that even when the ball is about to hit the opponent's goal, it mimics the actions of the ball almost exactly.

It is likely that the network associated the position of the opposing AI's paddle with the reward of scoring a point. This means that when the goal was scored, and the network's paddle was in the same position as the AI's, it was associated with the positive reward.

## VI. CONCLUSION

From the overall experimentation, certain features of video games were shown to have a positive effect on training the AI. One of the main features was giving rewards that immediately corresponded with correct actions in the game. Examples of this would be in Ms. Pac-Man where inputs corresponded to increases of score due to eating pellets. In games where the inputs did not correspond to an immediate reward, such as Space Invaders and Breakout, the results were much more mixed. This is also closely tied with the number of rewards that are available, as games with sparse rewards trained more poorly and had more erratic behaviour.

A second feature is the "default" state of the game, and the ability of the inputs to affect the outcome. Pinball, for example, trained poorly, due to inputs not having an effect, except for very specific circumstances (when the ball is near the paddle). Conversely, Ms. Pac-Man and Qbert have scenarios where almost any input leads to an increase of score, making them easier to train to perform effective actions.

As evidenced by the Pong game, some indication of correct behaviour may also be useful to train an AI, rather than needing to program specific rewards. With the AI associating the opponent's paddle with a reward, it unintentionally learned to play the game by following the paddle, and by extension the ball.

A negative feature of a game comes from Asteroids, where inputs have lingering effects and more long-term reasoning is required to avoid obstacles. The network would require knowledge of the current speed and direction of the ship to input commands to avoid asteroids, and also be able to competently pilot the ship. This also ties into the "default" state of the game, where giving no inputs can lead to a longer playtime and therefore more score, whereas pressing inputs is more likely to cause the agent to fail.

Another negative feature is the amount of randomness in the game. Most of the games have a degree of randomness in them, but video games with a higher degree of randomness trained worse than those with more fixed structures. This can be seen with Asteroids and Pinball, where actions outside of player control have large effects on game state (e.g.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 7, July 2019

initial position and direction of asteroids). This makes it more likely that the agent will be unable to make an effective decision based on its previous inputs.

Overall, the features found were demonstrated in video games, but could also be applied to other reinforcement based neural network task. For exercises in more efficiently training an AI, adjusting rewards is a large part of the process, which can be made easier or harder depending on the game's features, as well as changing what information is presented to the AI. While positive features can make training easier, negative features can also be mitigated by proper pre-processing of data, longer training times, and modifications to the network. However, these add additional overhead onto the training process and need to be properly applied to avoid negative effects. In addition, the challenge of not knowing exactly how the network came to perform a specific action is one of the main limiting factors in determining features and effective techniques for training.

## REFERENCES

- [1] A. A. Rusu *et al.*, 'Progressive Neural Networks', *arXiv:1606.04671 [cs]*, Jun. 2016.
- [2] X. Glorot and Y. Bengio, 'Understanding the difficulty of training deep feedforward neural networks', in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
- [3] C. Chen, A. Seff, A. Kornhauer, and J. Xiao, 'DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving', 2015, pp. 2722–2730.
- [4] M. Mueller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, 'UE4Sim: A Photo-Realistic Simulator for Computer Vision Applications', Aug. 2017.
- [5] D. Silver *et al.*, 'Mastering the game of Go with deep neural networks and tree search', *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.
- [6] P. Liskowski, W. M. Jaskowski, and K. Krawiec, 'Learning to Play Othello with Deep Neural Networks', *IEEE Transactions on Games*, pp. 1–1, 2018.
- [7] D. Silver *et al.*, 'Mastering the game of Go without human knowledge', *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.
- [8] A. Kalyanpur and M. Simon, 'Pacman using genetic algorithms and neural networks', *University of Maryland*, 2001.
- [9] S. M. Lucas, 'Evolving a Neural Network Location Evaluator to Play Ms. Pac-Man', in *Proceedings of the 2005 IEEE Symposium on Computational Intelligence and Games (CIG 2005)*, 2005, pp. 203–210.
- [10] H. van Seijen, M. Fatemi, J. Romoff, R. Laroche, T. Barnes, and J. Tsang, 'Hybrid Reward Architecture for Reinforcement Learning', *arXiv:1706.04208 [cs]*, Jun. 2017.
- [11] L. Galway, D. Charles, and M. Black, 'Machine learning in digital games: a survey', *Artif Intell Rev*, vol. 29, no. 2, pp. 123–161, Apr. 2008.
- [12] J. M. F. Fernandez and T. Mahlmann, 'The Dota 2 Bot Competition', *IEEE Transactions on Games*, pp. 1–1, 2019.
- [13] K. Grace, J. Salvatier, A. Dafoe, B. Zhang, and O. Evans, 'Viewpoint: When Will AI Exceed Human Performance? Evidence from AI Experts', *I*, vol. 62, pp. 729–754, Jul. 2018.
- [14] A. Garriga Alonso, 'Solving Montezuma's Revenge with Planning and Reinforcement Learning', 2017.
- [15] N. Justesen, P. Bontrager, J. Togelius, and S. Risi, 'Deep Learning for Video Game Playing', *arXiv:1708.07902 [cs]*, Aug. 2017.
- [16] X. Guo, S. Singh, H. Lee, R. L. Lewis, and X. Wang, 'Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning', in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3338–3346.