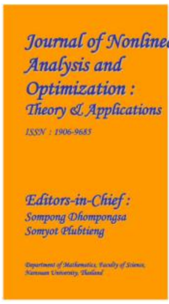
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.1 : 2024
ISSN : **1906-9685**



EXPLORATION OF SCRABBLE GAME IMPLEMENTATION THROUGH NASH EQUILIBRIUM AND U NET ARCHITECTURE

Dr. Alok Singh Gahlot, Asst Prof, Department of Computer Science & Engg. MBM University, Jodhpur, aloksg@gmail.com
Ruchi Vyas, Ph.D. Scholar, Department of Computer Science & Engg. MBM University, Jodhpur, ruchivyas85@gmail.com

Abstract
This research explores the convergence of game theory and deep learning in the context of Scrabble, aiming to revolutionize the implementation of intelligent board games. Leveraging the principles of Nash Equilibrium, a foundational concept in game theory, we investigate the strategic decision-making dynamics in Scrabble gameplay. Our objective is to achieve a balance that enhances player experiences by identifying stable strategies for optimal moves and word placements, ultimately maximizing point gains. The fusion of Nash Equilibrium and the U-Net architecture represents a pioneering effort to bridge the gap between strategic decision-making and advanced machine learning [1]. By synergizing these elements, our exploration seeks to unravel new dimensions in Scrabble gameplay, offering not only challenging adversaries for players but also an adaptive and engaging gaming experience. The proposed framework aims to contribute to the broader discourse on intelligent gaming, presenting a comprehensive and innovative perspective on the amalgamation of game theory and deep learning in the enhancement of classic board games like Scrabble.
Keywords: Scrabble, U-Net architecture, Game theory

I. INTRODUCTION

In the realm of board games, Scrabble stands out as a classic that challenges players to strategically arrange letters into words on a grid, earning points based on the complexity and length of their creations. As technology advances, the integration of artificial intelligence (AI) and deep learning techniques has opened new avenues for enhancing gameplay experiences. This research embarks on a journey to explore the implementation of the venerable Scrabble game, leveraging the synergistic power of Nash Equilibrium and the innovative Deep Learning U-Net architecture [2]. Scrabble, known for its blend of vocabulary skills and strategic thinking, provides an intriguing domain to investigate the application of game theory principles. Nash Equilibrium, a foundational concept in game theory, focuses on finding stable strategies in competitive situations where each participant aims to maximize their gains. In the context of Scrabble, the pursuit of optimal moves and strategic word placements aligns seamlessly with the principles of Nash Equilibrium, fostering a balance that enhances gameplay dynamics. Deep Learning U-Net architecture, renowned for its success in image segmentation tasks, presents a unique and unexplored avenue for the implementation of Scrabble. By applying the principles of the U-Net architecture to the context of the game, we aim to develop a system that not only understands the linguistic intricacies of word formation but also optimally navigates the Scrabble board, strategically placing tiles to achieve maximum point gain. The fusion of Nash Equilibrium and the U-Net architecture in Scrabble implementation promises a novel approach to elevate the player experience. This research delves into the intricate interplay between strategic decision-making and deep learning algorithms, pushing the boundaries of what is achievable in the realm of intelligent gaming. Through this exploration, we seek to unravel the potential synergies that arise when the elegance of Nash Equilibrium converges with the computational prowess of Deep Learning U-Net, unlocking a new dimension of sophistication in Scrabble gameplay.

II. LITERATURE REVIEW

The application of game theory in board games has been a subject of extensive research, with scholars exploring strategic decision-making, optimal moves, and equilibrium strategies. In the context of Scrabble, studies have delved into player interactions, strategies, and the quest for Nash Equilibrium to achieve balanced and stable gameplay dynamics [3]. Nash Equilibrium, as introduced by John Nash, has been a cornerstone in game theory, providing a theoretical framework to understand and predict rational decision-making in competitive situations [4]. Research has explored its applicability in various domains, and its potential impact on board games, including Scrabble, where players aim to maximize their points while considering the moves of their opponents. The integration of deep learning techniques in the gaming domain has witnessed significant advancements. Research on convolutional neural networks (CNNs) and architectures like U-Net has demonstrated success in image recognition, segmentation, and pattern analysis [5]. However, the utilization of these techniques in the context of board games, particularly Scrabble, remains relatively unexplored. Scrabble's core challenge lies in the manipulation of language, making natural language processing (NLP) an essential consideration. Studies on word embeddings, such as Word2Vec and GloVe, have explored how deep learning can capture semantic relationships between words. These embeddings can potentially enhance the language understanding component of Scrabble gameplay. While existing research has touched upon AI implementations in Scrabble, the incorporation of both Nash Equilibrium and deep learning U-Net architecture in the same framework remains a unique and underexplored endeavour. The literature offers insights into individual components but lacks a comprehensive exploration of their combined impact on the strategic dynamics of Scrabble.[6][7]

III. SCRABBLE GAME

Standard board of scrabble has a large square shape with 225 cells of height and width 15 * 15 to accommodate tile of letters into each cell. A bag of tiles with quantity 100 is used to play the scrabble game where 98 tiles have letter on it and 2 tiles are blank which can be used with any substituted letter. All tiles have a point value on the bottom to ensure their values that could be used to calculate the total sum of word apart from blank tiles they do not have any point and it is always 0.

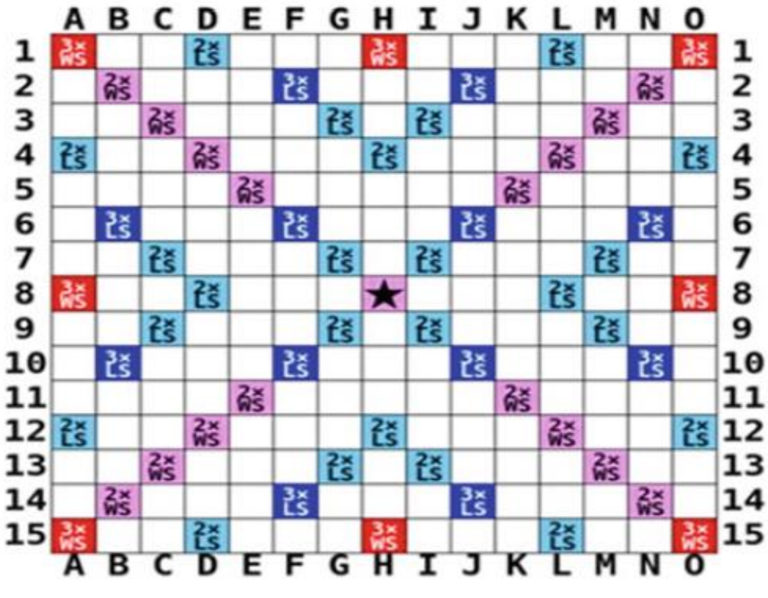


Figure 1: Scrabble Board

Tiles with letter s, r, a, e, i, o, u, n, l has score value point 1, Letter g and d has score point Value as 2. The Letters m, p, c, b has score point 3, The letters h, y, v, w, f has score point as 4, Letter k with points 5, x and j with points 8 and z and q with points 10. Scrabble board has certain cells with extra

score which multiplies the letter score or word score which eventually increases the score at its peak. Such cells include multipliers Double letter, Double Word, Triple letter, Triple Word which increases the word point sum of a player. Placing at theses cell position adds on to the score and may increases the chances of winning the game for a player. These special multiplier cells are painted with different colour so that they could be easily identified and has their unique significance. Double letter multiplier just doubles the score of letter placed on the cell. Double word multiplier cell doubles the complete score of word made which has included that cell. Triple letter multiplier increases the score of letter thrice placed on the cell. Triple word multiplier cell increases the complete score of word by three times.[8]
Cells with double letter, double word, triple letter, triple word have a clause on their usage as if the player has used a multiplier cell for a word, then it cannot be used for extra points for the next word. Players will take one tile out of the bag without looking at any of the other tiles. The game will start for the player whose letter is closest to "A." At the beginning of the game, the blank tile will win. After that, the tiles are put back in the bag and utilised for the rest of the game. To begin their turn, each player will draw seven tiles out of the Scrabble bag. In every round, there are three possible outcomes. The options available to the player are to insert a word, swap out tiles for fresh ones, or pass. As the other two possibilities do not result in a point, players will often attempt to put a word. A player can exchange any or all the tiles they presently own when they decide to trade tiles. The players' turn ends when the tiles are traded, and they cannot place a word on the board until their subsequent turn.[9]
It is always an option for players to pass. They will give up that turn and hope to get another chance to play. The player with the highest score wins when the game ends and no player pass twice in a succession. When the game begins, the first player will place their word on the star spin in the centre of the board. The star is a double square and will offer a double word score. All players following will build their words off this word, extending the game to other squares on the board. Players will draw fresh tiles to cover those that are lost when they are played on the board. Throughout the game, players will always have seven tiles. To ensure that the letters are always a mystery, drawing tiles is always done without seeing into the bag. The player who uses all 7 tiles for a word on a scrabble board the player will receive a 50 points bonus with the total sum points of the word made.When all the tiles are placed on the board and the bag of tile is empty, the game ends and the player with maximum reward points will win the game. The score points are being tallied and calculated to find score of each player as scrabble game came can have two to four players. If the players are left with certain tiles and cannot be played on the board then those tile points are added and they are being deducted from their actual point sum and then player with highest score is declared winner [10].

IV. DEEP LEARNING ARCHITECTURE
Deep learning techniques, particularly neural networks, have been employed in Scrabble AI. Neural networks can be trained to predict the value of different moves, considering various features and patterns. Deep learning models bring the advantage of automatically learning relevant features from the data, potentially capturing intricate relationships between moves and outcomes.[9]

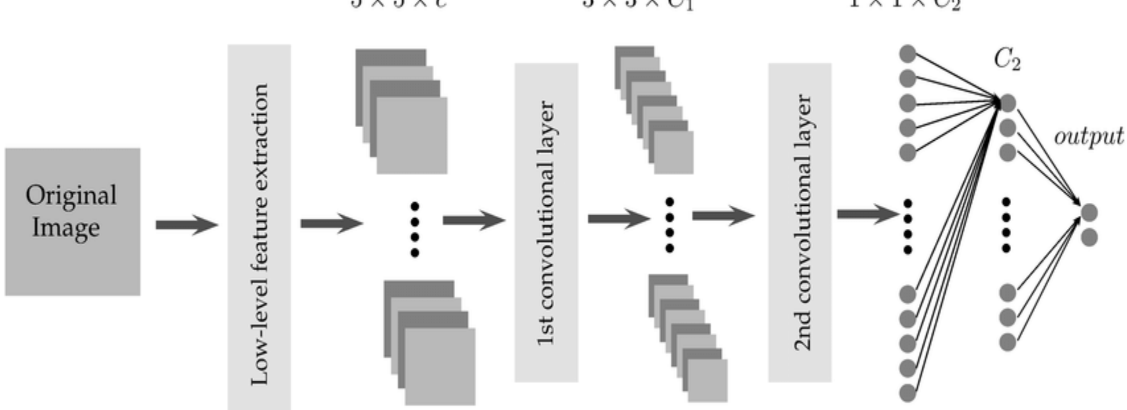


Figure 2: An example showing Deep learning implementation

V. PROPOSED ALGORITHM

The algorithm incorporates an "Autoencoder Training" step, where, if the move improves the autoencoder's ability to represent the board, the model is retrained with the updated data. Game Theory considerations are implemented in the subsequent step, evaluating decision-making based on opponent moves, future potential moves, and strategic tile placements. The "End-of-Turn" step checks for the end of the game or transitions to the next player's turn, looping until the game is concluded. The end of the game involves calculating final scores and declaring a winner.If desired, the autoencoder undergoes additional training with the entire game history in the "Autoencoder Training" phase to enhance its ability to represent diverse board states. Finally, the algorithm concludes with a "Conclusion" step, summarizing game results and key strategies employed, along with reflections on the effectiveness of integrating the autoencoder and game theory.

Step 1. Initialization: Set up the Scrabble board and initialize the autoencoder model for board representation also define the scoring system.

Step 2. Game Loop

while game_not_over

2.1 Board Representation Encode the current state of the Scrabble board using the autoencoder and obtain a compressed representation of the board.

2.2 Word Suggestion: Use the autoencoder representation to suggest potential word placements and leverage the encoded features to predict suitable positions for new tiles.

2.3 Player Move: Allow the player to make a move, Validate the move against the rules.

2.4 Update Board: Update the Scrabble board with the player's move.

2.5 Scoring: Calculate the score based on the updated board state.

2.6 Autoencoder Training: If the move improves the autoencoder's ability to represent the board, retrain the autoencoder with the updated data.

2.7 Game Theory Considerations: Implement game theory principles for decision making. Consider opponent moves, future potential moves, and strategic tile placements.

2.8 End-of-Turn: Check for the end of the game or move to the next player's turn.

End

Step 3. End of Game: Calculate the final scores and declare the winner.

Step 4. Autoencoder Training: If desired, retrain the autoencoder with the entire game history to improve its ability to represent different board states.

Step 5. Conclusion: Summarize the game results and key strategies employed and Reflects on the effectiveness of the autoencoder and game theory integration.

VI. RESULTS

The analysis of trained network is done and below is the graph between RSME Values iteration and Loss function value and iterations.

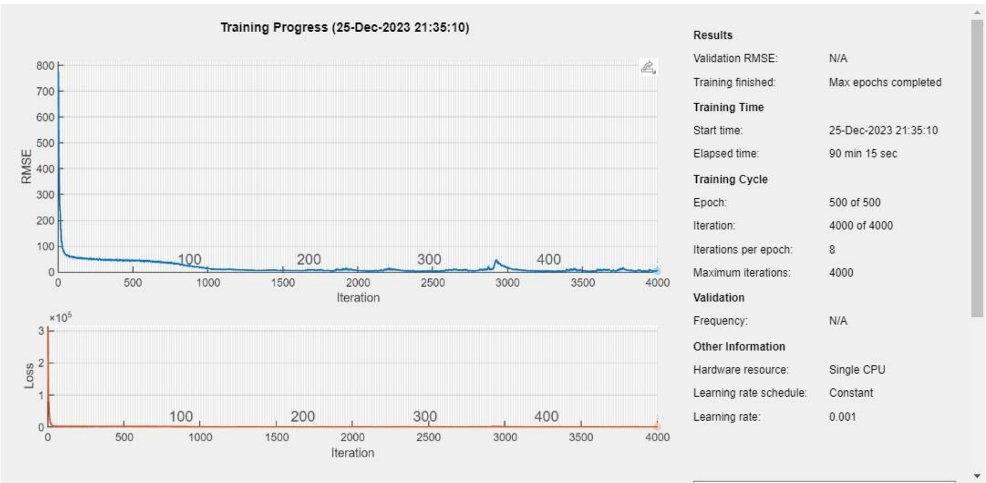


Figure 3 Graph Shows RSME Values and Loss function Value

```
xTrainingData      yDesiredOutPut     oActualTNOutPut
SVLRUUI#########   SVLRUU.#########   SVLRUU.#########
...............#   ...............#   ...............#
...............#   ...............#   ...............#
...............#   ...............#   ...............#
...............#   ...............#   ...............#
...............#   ...............#   ...............#
......A....G...#   ......A....G...#   ......A....G...#
......R..R.END.#   ......R..R.END.#   ......R..R.END.#
.....PET.EAT...#   .....PET.EAT...#   .....PET.EAT...#
.......RED...O.#   .......RED...O.#   .......RED...O.#
.....ONE...O.U.#   .....ONE...O.U.#   .....ONE...O.U.#
.......END.NOT.#   .......END.NOT.#   .......END.NOT.#
.....HIS.AGE...#   .....HIS.AGE...#   .....HIS.AGE...#
.....I..MY.....#   .....I..MY.....#   .....I..MY.....#
....IS.........#   ....IS.........#   ....IS.........#
....F..........#   ...IF..........#   ...IF..........#
```

Figure 4 Shows Sample output after training Scrabble board with desired output and actual output

Table 1 Shows Data for 500 Epochs

```
Training on single CPU.
Initializing input data normalization.
|======================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |     RMSE     |     Loss     |      Rate       |
|======================================================================================|
|       1 |           1 |       00:00:05 |       775.05 |     300349.3 |          0.0010 |
|       7 |          50 |       00:01:03 |        66.79 |       2230.7 |          0.0010 |
|      13 |         100 |       00:01:59 |        59.52 |       1771.1 |          0.0010 |
|      19 |         150 |       00:02:56 |        55.17 |       1521.8 |          0.0010 |
|      25 |         200 |       00:03:53 |        47.91 |       1147.9 |          0.0010 |
|      32 |         250 |       00:04:33 |        49.95 |       1247.3 |          0.0010 |
|      38 |         300 |       00:05:03 |        49.31 |       1215.8 |          0.0010 |
|      44 |         350 |       00:05:36 |        50.21 |       1260.8 |          0.0010 |
|      50 |         400 |       00:06:33 |        49.11 |       1205.7 |          0.0010 |
|      57 |         450 |       00:07:30 |        47.08 |       1108.3 |          0.0010 |
|      63 |         500 |       00:08:28 |        41.25 |        850.7 |          0.0010 |
|     425 |        3400 |       01:18:52 |         6.65 |         22.1 |          0.0010 |
|     432 |        3450 |       01:19:53 |        14.22 |        101.1 |          0.0010 |
|     438 |        3500 |       01:20:55 |         4.90 |         12.0 |          0.0010 |
|     444 |        3550 |       01:21:44 |         2.13 |          2.3 |          0.0010 |
|     450 |        3600 |       01:22:35 |         7.63 |         29.1 |          0.0010 |
|     457 |        3650 |       01:23:32 |         6.04 |         18.2 |          0.0010 |
|     463 |        3700 |       01:24:21 |         6.23 |         19.4 |          0.0010 |
|     469 |        3750 |       01:25:15 |        15.52 |        120.5 |          0.0010 |
|     475 |        3800 |       01:26:31 |         5.34 |         14.3 |          0.0010 |
|     482 |        3850 |       01:27:26 |         3.03 |          4.6 |          0.0010 |
|     488 |        3900 |       01:28:21 |         2.89 |          4.2 |          0.0010 |
|     494 |        3950 |       01:29:13 |         7.10 |         25.2 |          0.0010 |
|     500 |        4000 |       01:30:15 |         2.87 |          4.1 |          0.0010 |
|======================================================================================|
Training finished: Max epochs completed.

 Trained Network file: .\output\tNet_S_0000106_E_0000500_I_0004000__2023_12_25_23_05_28.mat
 xTrainingData:   | yDesiredOutPut:   | oActualTNOutPut:
```

VII. CONCLUSION & FUTURE WORK

We embarked on a journey to revolutionize Scrabble gameplay by merging cutting-edge technologies from machine learning and game theory. The autoencoder complemented this by capturing latent features and nuances, further refining the algorithm's understanding of complex board configurations and strategic possibilities. The integration of Nash Equilibrium added a dynamic strategic layer,

allowing the algorithm to respond intelligently to opponent moves and adjust its decision-making strategy accordingly.
Further refinement and exploration of deep learning architectures can be undertaken to improve the algorithm's understanding of nuanced game scenarios. The incorporation of more sophisticated neural network structures may yield even more accurate and adaptive decision-making.

REFERENCES
1. A. A. A. Abass, N. B. Mandayam and Z. Gajic, "Evolutionary Random-Access Game with Objective and Subjective Players," in IEEE Access, vol. 9, pp. 35562-35572, 2021, doi: 10.1109/ACCESS.2021.3062212.
2. N. Karunanayake, A. Wijesinghe, C. Wijethunga, C. Kumaradasa, P. Jayasekara and R. Rodrigo, "Towards a Smart Opponent for Board Games: Learning beyond Simulations, "IEEE 2020.
3. X. Jin, K. Li, Q. -S. Jia, H. Xia, Y. Bai and D. Ren, "A Game-Theoretic Reinforcement Learning Approach for Adaptive Interaction at Intersections," 2020 IEEE
4. M. A. Samsuden, N. M. Diah and N. A. Rahman, "A Review Paper on Implementing Reinforcement Learning Technique in Optimising Games Performance," Shah Alam, Malaysia, 2019, pp. 258-263, IEEE
5. Aung, H.P.P.; Khalid,M.N.A.; Iida, H. What Constitutes Fairness in Games? A Case Study with Scrabble. Information 2021, 12, 352. https://doi.org/10.3390/info12090352
6. K. Arulkumaran, M. P. Deisenroth, M. Brundage and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," in IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26-38, Nov. 2017
7. M. C. Fu, "AlphaGo and Monte Carlo tree search: The simulation optimization perspective," 2016 Winter Simulation Conference (WSC), Washington, DC, USA, 2016, pp. 659-670, IEEE.
8. Amit Benbassat and Moshe Sipper. Evo MCTS: Enhancing MCTS-Based Players through Genetic Programming. In Proc. IEEE Conf. Comput. Intell. Games, pages 57–64, Niagara Falls, Ontario, Canada, 2013.
9. Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems".
10. Li Zhang, Wei Wang, Shijian Li, Gang Pan. (2019) "Monte Carlo Neural Fictitious Self-Play: Approach to Approximate Nash equilibrium of Imperfect-Information Games".