CS 686 Project Proposal: Mortal Kombat AI

Hussein Saafan

June 4, 2022

Historically, games have provided a controlled setting for showcasing AI techniques. This includes games like chess [1], checkers [2], Go [3], and more recently video games such as Star Craft II [4]. Mortal Kombat is a 1992 DOS fighting game developed by Midway games. It is easy to learn with only 9 controls but difficult to master. The game includes an arcade ladder mode and a 2 player vs mode with 7 playable characters. The aim of this project is to create an AI capable of beating the arcade ladder.

The first challenge is to determine what is happening on screen which would require the application of computer vision techniques such as object detection [5] and action recognition[6]. YOLOv5 [7] is an efficient object detection library that would be useful for this real time application. Alternatively, since the game uses sprites, search techniques could be used to match pixels on screen to a sprite library.

The second challenge is to be able perform combos. To do this, the AI must be aware of past actions and plan ahead as well as take into account character positioning. This can be tackled using techniques such as recurrent neural networks or long short-term memory (LSTM) networks [8]. A deep LSTM was used in AlphaStar [4]. Since characters have different move sets, transfer learning [9] could be used to train the AI to play different characters.

One way to provide feedback to the AI is to use the score or character health as a metric of performance. Another way is to use the number of wins and losses. While wins are the ultimate goal, using them as a metric would require making predictions about future rewards to update network weights. This can be done using methods like true online temporal-difference training [10]. The other two metrics can be periodically sampled to update the network. Character recognition [11] can be used to extract the score or detect win/loss messages. The health can be extracted using the position of red/green pixels of a fixed screen section.

The AI can be trained by playing against itself. Self play was used in AlphaGo Zero [3] with great success but Vinyals et al. noted that using self play exclusively led to "chas[ing] cycles in [the] strategy space" [4]. Another method that would prevent cyclic strategies is training on the in game ladder mode. The last option is to have a human play against the AI.

Other tasks include detecting the current scene to determine what set of actions to take. For example, selecting a character on the character select screen, resting between matches, and fighting. The start and end of fights can be detected using the same character recognition mentioned above and a similarity metric for images can be used for static scenes such as character selection. In addition to these challenges, a framework must also be constructed to allow the AI to interact with the game. Depending on resource availability, the easiest/hardest challenge is to design the AI efficiently to run in real time.

References

[1] F.-H. Hsu, *Behind Deep Blue: Building the computer that defeated the world chess champion.* Princeton University Press, 2002.

[2] J. Schaeffer, N. Burch, Y. Bjornsson, A. Kishimoto, M. Muller, R. Lake, P. Lu, and S. Sutphen, “Checkers is solved,” *science*, vol. 317, no. 5844, pp. 1518–1522, 2007.

[3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, “Mastering the game of go without human knowledge,” *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[4] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, “Grandmaster level in starcraft ii using multi-agent reinforcement learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[5] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, “Deep learning for generic object detection: A survey,” *International journal of computer vision*, vol. 128, no. 2, pp. 261–318, 2020.

[6] H.-B. Zhang, Y.-X. Zhang, B. Zhong, Q. Lei, L. Yang, J.-X. Du, and D.-S. Chen, “A comprehensive survey of vision-based human action recognition methods,” *Sensors*, vol. 19, no. 5, p. 1005, 2019.

[7] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, and M. T. Minh, “ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference,” feb 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6222936

[8] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” *arXiv preprint arXiv:1506.00019*, 2015.

[9] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[10] H. Van Seijen, A. R. Mahmood, P. M. Pilarski, M. C. Machado, and R. S. Sutton, “True online temporal-difference learning,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 5057–5096, 2016.

[11] N. Islam, Z. Islam, and N. Noor, “A survey on optical character recognition system,” *arXiv preprint arXiv:1710.05703*, 2017.