Replicating the Pros
Super Smash Bros. Melee Imitation Learning Agent

Ben Rodriguez-Vars
Professor Radev
CPSC 490: Senior Thesis
February 9, 2017
The goal of my project is to create a learning agent that is capable of mimicking the play style and ability of a professional Super Smash Bros. Melee player. I am an avid fan and player of the game, which sparked my interest in pursuing this goal. I've seen a video of a bot playing against various pros and absolutely destroying them. However, the bot was hardcoded to respond to certain states with certain actions against certain characters. The programmer who developed this bot mentioned wanting to build an AI that learned how to play the game from scratch, but he never seemed to post anything about it. I did find a thesis from MIT where a PhD student trained an agent to play melee at a competitive level by playing itself and gaining knowledge through reinforcement learning. This gave me more confidence that this project could be accomplished despite the complexity of the numerous states and actions. Except, this isn't exactly the same problem since it lacks the computer vision aspect that I am tackling. I found a more relevant paper, this one from PhD students at Stanford focusing on game imitation through video. They used convolution networks to train an agent on a couple hours of self-developed gameplay. The Stanford students did this with the Nintendo 64 version of Super Smash Bros.

While the project the Stanford students worked on is similar to the one I am pursuing, I am hoping to be able to train a bot strictly from video gameplay, which can prove to have potentially insurmountable problems. The Stanford students were able to train their agent because they were able to record their game data and feed that info alongside screenshots of what was going on in the game. This allowed the agent to learn which action to take in each state. Except, if I will be training the agent with just video from professional gameplay, I will not have access to their inputs and therefore
cannot use the method the Stanford students used. To be able to accomplish my goal I would need to train an agent on game screenshots so it can determine what is being input. That way it could translate a video of a game into player inputs. From there I could use a similar method of feeding the agent screenshots with accompanying inputs, so it can learn. While potentially manageable, for the agent to properly learn the input classifier would have to be extremely accurate, so the agent is learning the correct moves instead of a misclassified one. In order to create a classifier with that level of accuracy requires a massive amount of screenshots in unique positions. From any position on a map, a character can input 54 unique actions. Due to the thousands of positions a single character can be, plus the even greater amount of combinations that exist of opponent position relative to your character makes this problem very computationally heavy. Then to ensure accuracy you would need to get tons of screenshots of a certain move in a certain spot. Basically, I would have to manually gather and classify an insane amount of images which might not be accomplishable in any amount of time. Except, if we employ the method used by the Stanford students, where they used their own footage and controller inputs I could potentially be able to quickly create my own collection of classified images. Since I have the inputs and the screenshots I can correctly classify any move I make on the screen. If I play for long enough and focus on gathering unique states and actions I can assemble a decent amount of properly classified data which can be used to train my input classifier. (See figures below of controller and example screenshots which would be paired with actions)
**Figure 1:** A GameCube Controller with buttons labelled.

**Figure 2:** Screenshot of Fox performing a down air. The classifier should recognize this input as down on the analog and a pressed A button.

**Figure 3:** Screenshot of Fox performing a back air. The classifier should recognize this input as left on the analog and a pressed A button.
After creating an accurate input classifier, I will need to train my learning agent. To be able to learn to mimic the moves made by professionals I will feed my agent the screenshot from the pro’s game and their input produced by my classifier. This way the agent starts to figure out what action to perform in certain states. The issue with this is that to be able to train the agent, we need a lot of professional game footage. Not only that, but the footage has to be the same character matchup and on the same map as my classifier, otherwise it will incorrectly label actions. Because of these requirements, there might not be enough footage to properly train the agent on professional gameplay. If that happens to be the case, I might try reaching out to a professional player and asking them to record and send me gameplay. If that is not possible I will alter my goal to try to get the agent to be able to mimic my play style. Even though in that case I will have access to my game data, I will only feed it video and make use of my classifier to figure out the inputs. If my classifier proves to be too inaccurate at classifying my moves, I will resort to feeding the agent the visuals along with my game data to try to have it mimic my play style.

My overall goal is to have the agent mimic play style. However, mimicking the style of a professional player would be ideal. Plus, to be able to accomplish such a task I would learn more about creating a classifier and how to pair that with a learning agent, thereby empowering reinforcement learning with computer vision. If I find that one of my barriers is insurmountable, I will move onto the next strategy by eliminating the problem. I want to give each method a real shot before skipping the problem, however I do want to be able to produce an agent that can mimic someone’s play style, whether that be a professional’s or mine.
I will be working closely with another student, Trey Lachance, on this project, but our goals are separate. He will be working on developing an agent that learns to play SSBM through playing itself and reinforcement learning algorithms, whereas my agent will learn through mimicking the play of a person. The methods and problems for each are unique and will be accomplished separately. However, we will be able to help each other accomplish initial tasks like reading game data, determining success, selecting characters and maps, and other game related objectives. But when it comes to our individual goals, we will be completing our own work. Obviously it will be fun to compare our results and mention each others work in our final reports as well.
Sources


Recurrent Neural Network Project on Mario Kart -
https://www.youtube.com/watch?v=Ipi40cb_Rsl

Efficiency Reduction for Imitation Learning -

Melee Character Frame Data - https://smashboards.com/threads/stratocasters-hitbox-system-new-download-link.283973/

Mew2King’s Twitter Page (Pro Player) - https://twitter.com/MVG_Mew2King

Mango’s Twitter Page (Pro Player) - https://twitter.com/C9Mang0