Detecting an Enemy in a Screenshot

|  |  |  |
| --- | --- | --- |
| Luc Dupuy  Department of Computer Science  Ryerson University  Toronto, Canada  [luc.dupuy@ryerson.ca](mailto:luc.dupuy@ryerson.ca) | Adam Malczewski  Department of Computer Science  Ryerson University  Toronto, Canada  [malcz@ryerson.ca](mailto:malcz@ryerson.ca) | Raj Mistry  Department of Computer Science  Ryerson University  Toronto, Canada  [raj.mistry@ryerson.ca](mailto:raj.mistry@ryerson.ca) |

1. **Motivation**

With the ongoing growth of video games, comes an ongoing growth in cheating technologies. The use of cheats in modern multiplayer games is difficult because of implemented anti-cheats. By using object detection and image classification, we hope to detect an enemy by providing the machine learning model with screenshots of the game with and without an enemy player.

1. **Method**

Data would be generated ourselves, by using the Counter Strike: Global Offensive(CSGO) game and game engine to automate the generation and labeling of screenshots that do and don’t contain enemies in them. Then through the supervised learning approach of classification, we would apply an algorithm that will be able to make observations about the images it is fed, and categorize them based on the content of the image.

1. **Intended Experiments**

By using data sets generated by us, it will be possible to use them for training the algorithm. We can generate as many screenshots/images data sets as needed, so we can see how it succeeds with images it has never seen before.

**IV. Planning and Milestones**

1. Generating the dataset (October 10) - Adam
2. Choosing the features from the dataset(s) - Everyone

Formatting the selected features to prepare for the algorithm implementation. - Luc & Raj

Setting up a repository on github - Adam(October 18)

1. Write the algorithm (November 7) - Everyone
2. Training the algorithm - Adam & Luc

Failure Analysis (November 14) - Raj

1. Documenting the observations and analysis of the results (How accurate the algorithm is at predicting the image classifications) (November 29) - Everyone
2. Adjusting weights and biases as needed to further improve the output (December 6) - Everyone
3. Writing the report - Everyone

Make the video (December 13) - Luc