

AI-Gamer

Creating and training an ML model to create an AI that can play the game Rocket League

Marcel Pratikto

Introduction

The **purpose** of this project is to learn how to create, train, and apply AI to a real-world activity. The **standards** that I will be following are coding, design, and learning model. The **scope** of this project covers computer science and data science concepts of training a machine learning model. The result is a simple AI that will be able to play Rocket League. The definition of what that entails will be discussed in this requirements document.

Requirements

These are the must have requirements for the project:

1. The project shall create a dataset based on a player's playstyle in the game Rocket League
2. The project shall create an AI by training an ML model using the dataset mentioned above, which I will call "**AI-Gamer**"
3. The AI-Gamer shall predict the following outputs:
 - a. Left joystick's x and y directions
 - b. Right trigger intensity
 - c. Left trigger intensity
 - d. If buttons X, A, B, left bumper, or right bumpers are pressed
4. The AI-Gamer shall use a webcam as its visual input
5. The AI-Gamer shall utilize OpenCV to detect a ball in its visual input
6. The AI-Gamer shall control the game using a virtual gamepad

Stretch Requirements

These are the nice to have requirements for the project:

1. The AI-Gamer should use a multi output regressor machine learning model
2. The AI-Gamer could be able to hit the ball in the game Rocket League
3. The AI-Gamer could be able to end its own program after completing a game in Rocket League
4. The AI-Gamer should be able to play Rocket League without any user inputs, apart from the user starting the game

Design Overview

Workflow

1. Prepare the setup
 - a. The webcam must be on a tripod that is level with the top half of the computer screen
 - b. Face the webcam towards the computer screen, placing it about a foot away

2. Launch Rocket League
3. Choose the rookie level goalie training game mode
4. Pause Rocket League after launching the game mode
5. Run AI-Gamer.py

Resources

1. Your computer must have Rocket League installed because that is the game that will be played by the AI
2. OpenCV will employ the webcam to detect the ball in Rocket League
3. My program, AI-Gamer.py, will use the output from OpenCV to feed to the custom ML model that I will create to emulate my playstyle
4. The output from my ML model will activate the virtual gamepad script that will control Rocket League

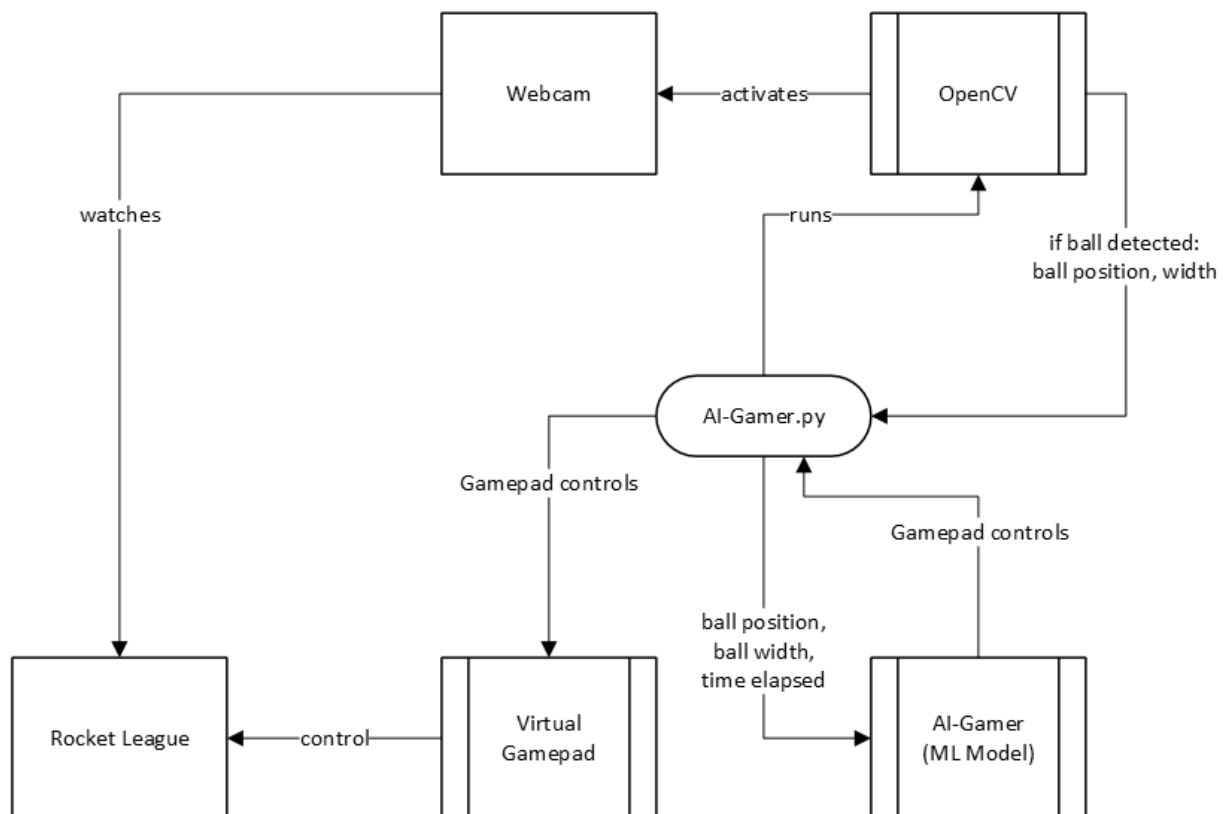
Data at Rest

No user information will be used by this program

Data on the Wire

No information will be saved by this program and no information will be transmitted

Data State



HMI/HCI/GUI

The following is a picture of the setup. This is what the user will see.



The following is a picture of what the AI will “see”. The AI-Gamer will use OpenCV to detect the ball in the game Rocket League.



Verification

Demo

The application works as designed if the AI will be able to play the game Rocket League without any user input. The user input would be the user having to start the AI-Gamer program after choosing the rookie goalie training game mode in Rocket League. It might work on other game modes, but this is the mode that it was trained for.

Testing

- Is the dataset valid enough for a machine learning model?
 - I will need to create a large enough dataset to train the ML model on.
 - If the csv file I create can be used to train an ML model, it will verify requirement 1.
- Can the ML model output the appropriate gamepad buttons / controls?
 - The hard part is figuring out what is going to be the inputs and outputs. Most machine learning models will only have one or two outputs, my ML model will need to have 9 outputs and at the time of writing this document there are only 5 inputs. How do I make sure that my inputs are valid enough to produce consistent outputs?
 - If the ML model can always produce 9 outputs, then it will verify requirements 2 and 3.
- Can the AI-Gamer program use OpenCV to detect the ball in Rocket League?
 - As my screenshots in the GUI section on the previous page show, if the program can “see” the ball, then it will verify requirements 4 and 5.
- Can the AI-Gamer move the player character in Rocket League?
 - If the AI moves the player character, then it will verify requirement 6. Bonus points if the character can hit the ball.

Sources

Gamepad

- <https://pypi.org/project/vgamepad/>

Keyboard

- <https://github.com/gauthsvenkat/pyKey>

Screen / switching between programs:

- https://www.reddit.com/r/learnpython/comments/xb10do/how_do_i_switch_between_open_programs_with_python/
- <https://pyautogui.readthedocs.io/en/latest/>

TensorFlow Object Detection:

- <https://automaticaddison.com/how-to-load-a-tensorflow-model-using-opencv/>
- <https://github.com/opencv/opencv/wiki/TensorFlow-Object-Detection-API>

Write to a csv file

- <https://www.geeksforgeeks.org/how-to-open-and-close-a-file-in-python/>
- <https://docs.python.org/3/library/csv.html>

Read and store gamepad inputs to a csv file

- <https://stackoverflow.com/questions/46506850/how-can-i-get-input-from-an-xbox-one-controller-in-python>
- <https://pypi.org/project/inputs/>

Save and load model

- https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/save_and_load.ipynb
- <https://www.geeksforgeeks.org/saving-a-machine-learning-model/>
- https://colab.research.google.com/gist/MarcelPratikto/d62275bac7d93282a885eb4e5c98717b/simple_nn_example.ipynb#scrollTo=w1jKaZouuVta

Multi Output Regressor Machine Learning Model

- <https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html>
- <https://stackoverflow.com/questions/57704609/multi-target-regression-using-scikit-learn>
- <https://medium.com/@tubelwj/developing-multi-class-regression-models-with-python-c8beca5dd482>
- <https://machinelearningmastery.com/multi-output-regression-models-with-python/>