Craft Prospect

Title: Neural Network-Based Arcade Game

Summary: Working as a team, the students’ goal is to create an arcade-style game which will be both educational and fun. The game’s purpose is to demonstrate how neural networks can be used to identify specific features in satellite imagery, a capability which Craft Prospect is developing in a CubeSat payload. By introducing this capability, satellites can immediately notify authorities of forest fires or illegal mining, tell farmers crucial information about their crops, and provide near-live tracking of ships across the globe. The students are given flexibility in how they present the use of neural networks and the format of the game: suggestions would be pitting the player against a trained neural network in a race to identify all of the features in a side-scrolling view of the Earth's surface. Requirements of the game are that it uses a neural network to label ground-based features, that the player interacts with the network somehow, that the benefits of the network in terms of identification speed or accelerated notification of relevant parties are clear, and that it's fun to play! The game must be playable in either a web app on the Craft Prospect website and/or as a standalone arcade unit powered by a Raspberry Pi or similar.

Resources:

2017 games for reference: https://craftprospect.com/media/games/

Simulation of orbital cloud detection, using the described payload: https://www.youtube.com/watch?v=YUQCB-nLmJE

Title: Satellite Demo Simulation

Summary: The goal of the project is to develop a high-quality simulation-driven animation for CubeSats in Earth orbit. Visualising satellite operations in space is both a useful tool during the development process and a crucial aspect of marketing products and services to customers and investors. In the supplied simulation video, the Python module VTK was used to provide a straightforward visualisation of a satellite in low Earth orbit which was driven by realistic data from a simulation of the satellite's orbital dynamics. Students will develop a more aesthetically pleasing visualisation tool, which similarly receives outputs from the simulation (such as latitude, longitude, orientation, sensor positions) and generates a realistic animation of the satellite in orbit. The animation does not need to be generated in real-time, but must be driven by the required data. Other requirements are that it must include realistic geolocating of the satellite above specific locations on Earth and be able to compile/render on computers with low-power graphics cards. Possible solutions include further enhancement of the VTK approach, augmentation of an existing tool such as STK (Satellite Toolkit) or use of Blender, which is known to integrate with Python. As well as developing a flexible visualisation frontend, the interface with Python must be adaptable and detailed comprehensively in software documentation.

Resources:

Simulation of orbital cloud detection, using Python and VTK: https://www.youtube.com/watch?v=YUQCB-nLmJE