Autonomous Model Car Optimized for Solar Energy Capture

Sarah Riseden

Dr. Stephen Medeiros PE

UCF College of Engineering and Computer Science

University of Central Florida

Duke Energy EXCEL Undergraduate Research Experience

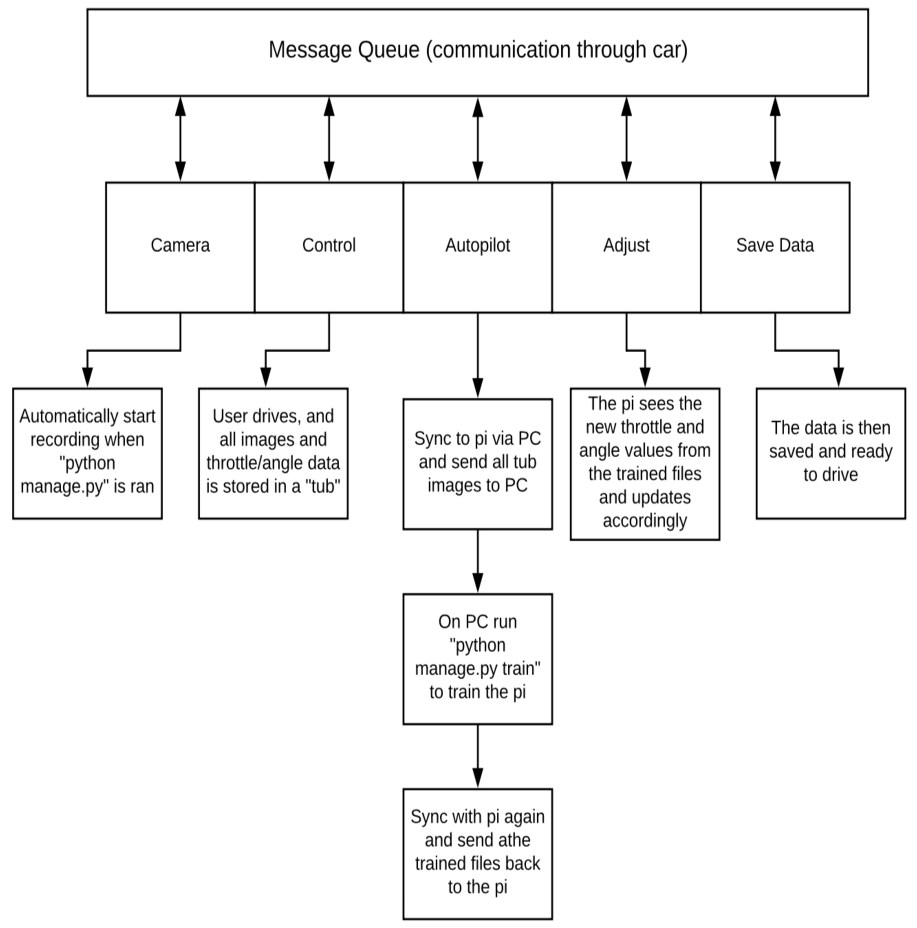
UCF

Abstract

The Donkey Car is an open source Python library that can add autonomous capability to an RC. The Donkey library utilizes machine learning modules including Tensorflow, Keras, OpenCV, and Tornado to create a small-scale autonomous vehicle. Using a Raspberry Pi, the RC car was re-wired to receive driving instructions from a smart phone or laptop via Wi-Fi or by a Bluetooth video game controller. While manually driving the car, the camera recorded the movements of the driver which were synchronized with the imagery captured by the Raspberry Pi camera. These data were utilized to train the autonomous driving system to replicate the actions of the driver. In Phase 2, a solar panel was integrated into the power supply of the car and its data was added to the machine learning training data stream on by the Raspberry Pi. The algorithm penalized entry into dark or low solar energy areas thus training the car to recognize areas of bright sunlight and to consider energy acquisition in its autonomous travel operations. The car autonomously found its own path in order to maximize energy capture alongside its general route objectives.

Background: Can a vehicle operate autonomously with a camera as its main sensor?

- Study tested neural networks using a small autonomous car powered by the "Donkey" library in Python.
- Built a model autonomous vehicle implementing image processing into neural networks.
- How can a machine be trained to react accordingly to various surroundings?
- How can the implementation of a solar panel assist the vehicle's driving algorithm?
- Future implementation of a solar panel to power the vehicle allowing the data from the solar panel to control the path of the car.
- Goal is to aid in future development of autonomous robotics



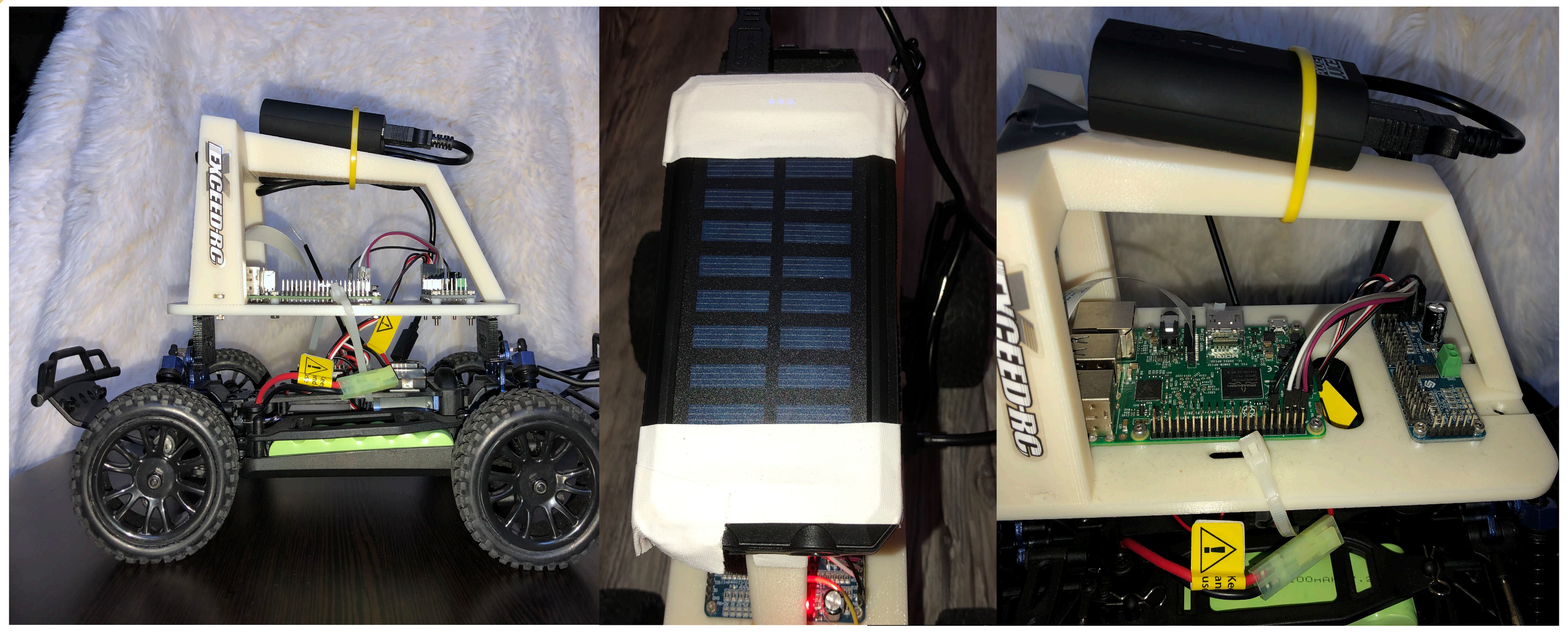


Figure 1: RC vehicle equipped with Raspberry Pi and solar panel

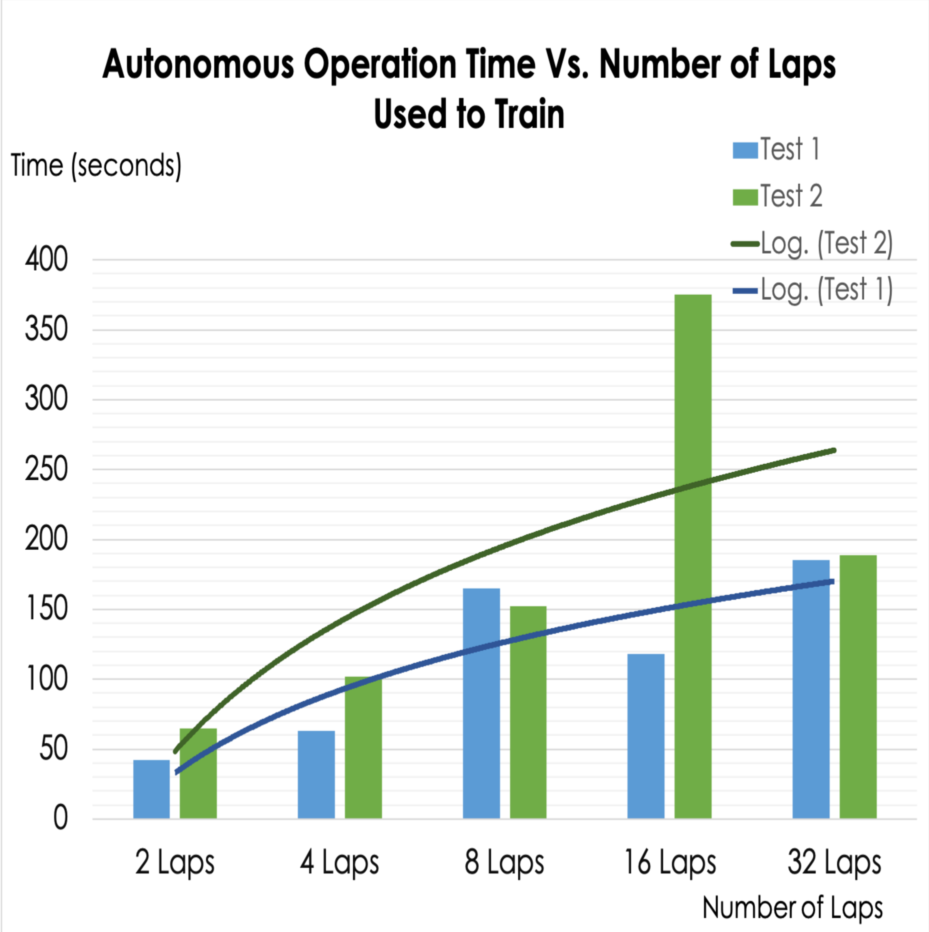


Figure 2: Recorded data from field tests

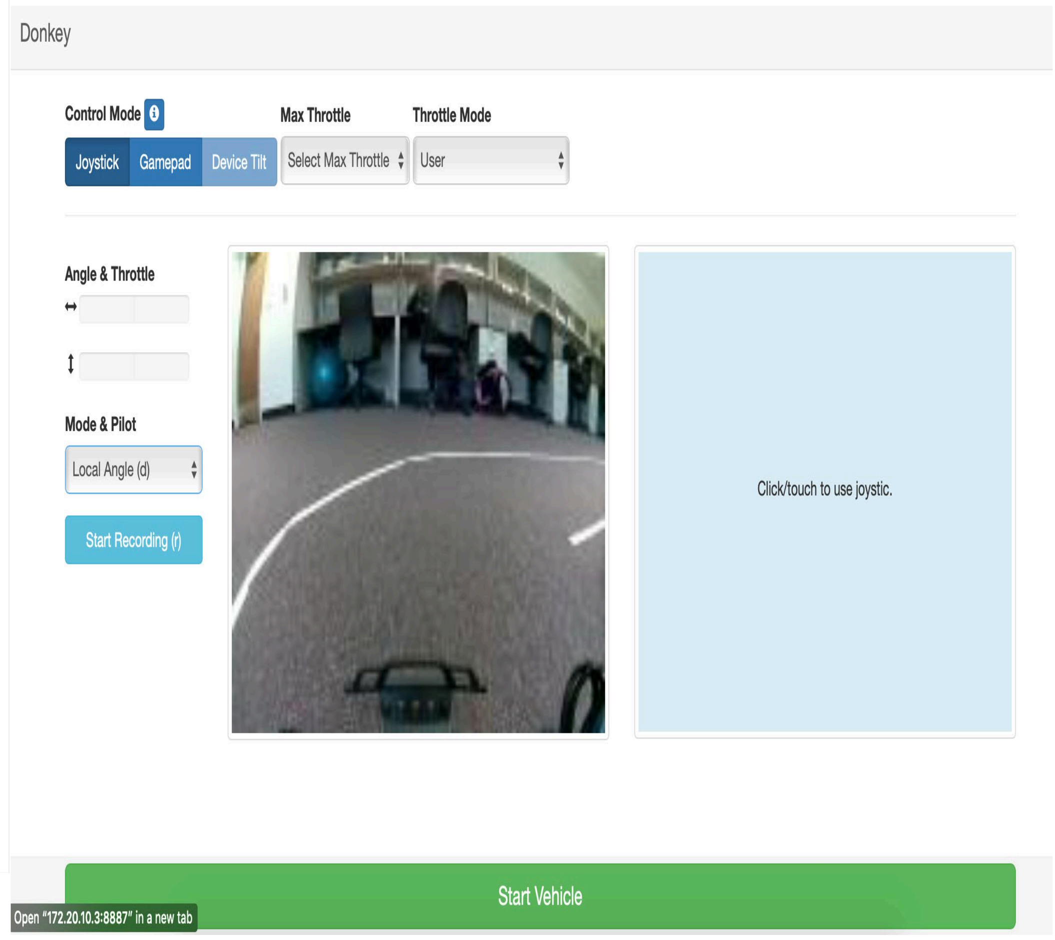


Figure 3: Donkey webserver

Figure 4: Test course and obstacle encountered

Methods and Results

- Various tests were done to determine if the car was behaving autonomously.
- After manually driving the car on a test track and collecting data, a training algorithm was run, and the car was set to steer autonomously on the same track, while speed was controlled manually.
- This was done on multiple different tracks in various settings, indoors and outdoors, to test the extent of the vehicle's training algorithms.
- On the control track, seen in Figure 4, tests were done to see if the number of training laps driven affected the autonomous operation of the vehicle.
- This was done to see if there is a cutoff point in which the training data sent to the vehicle becomes excess data, no longer assisting the operation.

Discussion And Future Research

- It was found that the car can perform in more controlled environments relying solely on vision.
- During testing the vehicle had a tendency to overshoot turns, which prevented further learning, as seen on Figure 2.
- Occasionally, if something in the background was of similar colors to track lines, the vehicle would veer off track at the point it saw this disrupt.
- If an obstacle was placed in the way of the path, the vehicle was seen to be almost "thinking" of a method to avoid it.
- While the vehicle wasn't specifically trained to avoid obstacles, it was able to avoid them in some cases, while in others, it collided with said obstacle.
- In the future, the car will be programmed to utilize the data stream from solar panels to maximize its energy efficiency while driving
- Potentially implementing solar streams into navigation systems can allow a driver to choose the best possible path based on efficiency maximizations.

References

- Chollet, F. (2015). Keras.
- Edward Murphy, Xiaohong Den. (2018). Donkey_Part_PS3Controller.
- Gurney, K. (1997). *An Introduction to Neural Networks*. University College London Press Limited, 11 New Fetter Lane, London, EC4P 4EE: Taylor & Francis Group.
- Kramer, T. (2017, 2018). PS3 Joystick Controller.
- Martín Abadi, A. A., Paul Barham, Eugene Brevdo,, Zhifeng Chen, C. C., Greg S. Corrado, Andy Davis,, Jeffrey Dean, M. D., Sanjay Ghemawat, Ian Goodfellow,, Andrew Harp, G. I., Michael Isard, Rafal Jozefowicz, Yangqing Jia,, Lukasz Kaiser, M. K., Josh Levenberg, Dan Mané, Mike Schuster,, Rajat Monga, S. M., Derek Murray, Chris Olah, Jonathon Shlens,, . . . Yuan Yu, a. X. Z. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Paper presented at the UNSENIX Symposium on Operationg Systems Design and Implementation, Savannah, GA, USA.
- Nitish Srivastava, G. H., Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research, 15*, 1929-1958.
- Roscoe, W. (2016, 2018). Donkey Car.

Acknowledgements

The author would like to thank Amr Oloufa and Omer Tatari for allowing use of their lab in Engineering Building 2, Room 180 and graduate students Burak Sen and Amirsaman Mahdavian for coordinating the schedule. The author would also like to thank Duke Energy for funding the project, and the EXCEL program at the University of Central Florida for their support and resources to complete this project.