# **KUBS**

**Robotic Car Follower with Image Recognition**

**From:**

Tonderai Kambarami

Luis Usseglio

Tarik Salay

Reed Bader

## **Abstract:**

This is the report of what we have worked on within this class IOT Robotics. It is a culmination of the efforts over the semester and serves as an example of what we went through for the last couple months. While it won’t be as informative as a research paper, it is hoped that nothing is missed from the work created. The project we worked on was a car that used visual image recognition software to discover objects around it and move toward them. The clear hope being that when it saw a person, that it would follow them, much like a robotic pet, which was accomplished in a rudimentary way, that can be improved upon if wanted. It was also meant to serve as a cheap gift to be made during the holidays. Something that could be bought and worked on, that would allow younger children to get a brief preview at the wonders of the computer science and robotics fields, since it can be done in a fun way that allows for bonding between parents and children.

## **Introduction:**

To begin, we were tasked with creating a project of our choice, and while we thought about this for a few days, going onto weeks, we thought of a good solution that fit with both the theme of the class, the spirit of the season and the general thought of our group. We wanted to create a robotic car that would be used as a kind of project for someone to buy their child, in the hopes that they would be able to spend time together and build a robotic car that used image recognition software. At first, we wanted to make it similar to a spy car, where it would be run from an app and would then be used to drive around the surrounding area and show images to the person controlling it. However, due to some complications we had while making the app and also a turn of thought halfway through, we decided on another course. Instead of a spy car, we wanted to make it like a robot pet, something that would chase down objects and follow them. It would be an interesting talking point and be fun it managed correctly. We managed to get it working, but there were still several issues.

## **Background Study / Literary Review**

When i came to background, we didn’t really look at too many resources, as we had a general idea of what we wanted to do. One of the main ones that we used as inspiration for our project, were the image recognition and robotic car ICP, since they gave us valuable resources to continue our project with. In all, we used resources provided in class and went with what we knew, rather than bite off more than we could handle at any given time.

## **Methodology**

Since it was difficult to acquire research or to find things that we wanted for our current type of project, we don’t have much in the way of our methodology. When we researched the project, we searched for specific pieces that would fit our ideals, such as:

For the car:

* A software that would be used to move the car around based on the inputs of an outside stimulus.
* A kind of car that would be cheap and affordable to use
* Something sturdier so that it is not easy to break
* Something that would be able to work with an arduino or Raspberry Pi

For the code:

* An image recognition software. The refresh rate for the camera to video was not overly important, only what was sent to the unit.
* Something that would not be overburdened by movement
* Something that would not burden the code or car with constant movement to try and keep going to new objects

## **Results and Evaluations**

The results were pretty simple to view, as we worked on it, we slowly got it more into working order, starting with the two (car and image recognition) as separate entities, as that was how building the two up would grow the quickest. After we got both to working order and out them together, we celebrated when it recognized things and started moving toward them. While we were saddened by the fact that we could not create a way for the car to be run without the use of a laptop hooked to it, as that was how we put it together, it still marked a success to us and we were happy to take it. All in all, it is not the most polished idea, and could use some more work to become what it was meant to be, but for something that we put time and effort in, it was nice to watch it pay off in the end and sparks ideas for the future.

## **Source Code**

The source code link is shown below:

[Link to source code in Git-Hub](https://github.com/luiper79/CS-490-5-Iot-Robotics/blob/master/PROJECT/SourceCode/SourCodeLAB2.txt)

## **Circuit Connections**



## **Project Preview and Images**

## **Project Video Links**

[Please click link to open video 1 in Vimeo](https://vimeo.com/378119629)

[Please click link to open video 2 in Vimeo](https://vimeo.com/378123072)

## **Conclusion and Future work**

In conclusion, the Kubs robotic car was not fully what we wanted, which is a bit frustrating, but it is not something that we will be put down by. We will grow better from the experience, and take away some ideas from this to possibly use in future projects. But as of right now, we do not have a desire or want to continue to improve the car, as we will be leaving and going down diverging paths, and may not see each other again. In terms of future work, we do not believe we have any here, but in our futures, we hope to have projects as interesting as this, but we have nothing planned for right now.

## **References**

EdjeElectronics. “EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi.” *GitHub*, 6 Nov. 2019,

<https://github.com/EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi.>

“OpenCV.” *OpenCV*, 14 Oct. 2019, <https://opencv.org/.>

Rosebrock, Adrian. “Raspberry Pi: Deep Learning Object Detection with OpenCV.” *PyImageSearch*, 25 Nov. 2018, <https://www.pyimagesearch.com/2017/10/16/raspberry-pi-deep-learning-object-detection-with-opencv/.>