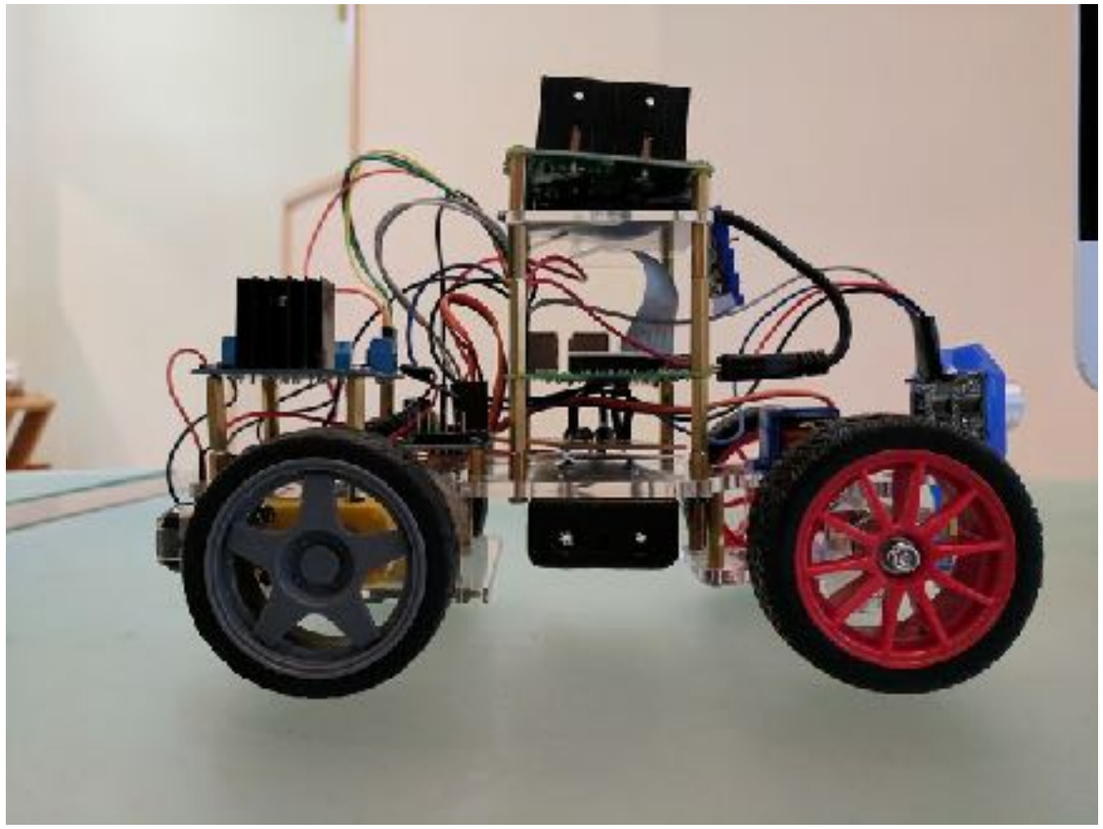
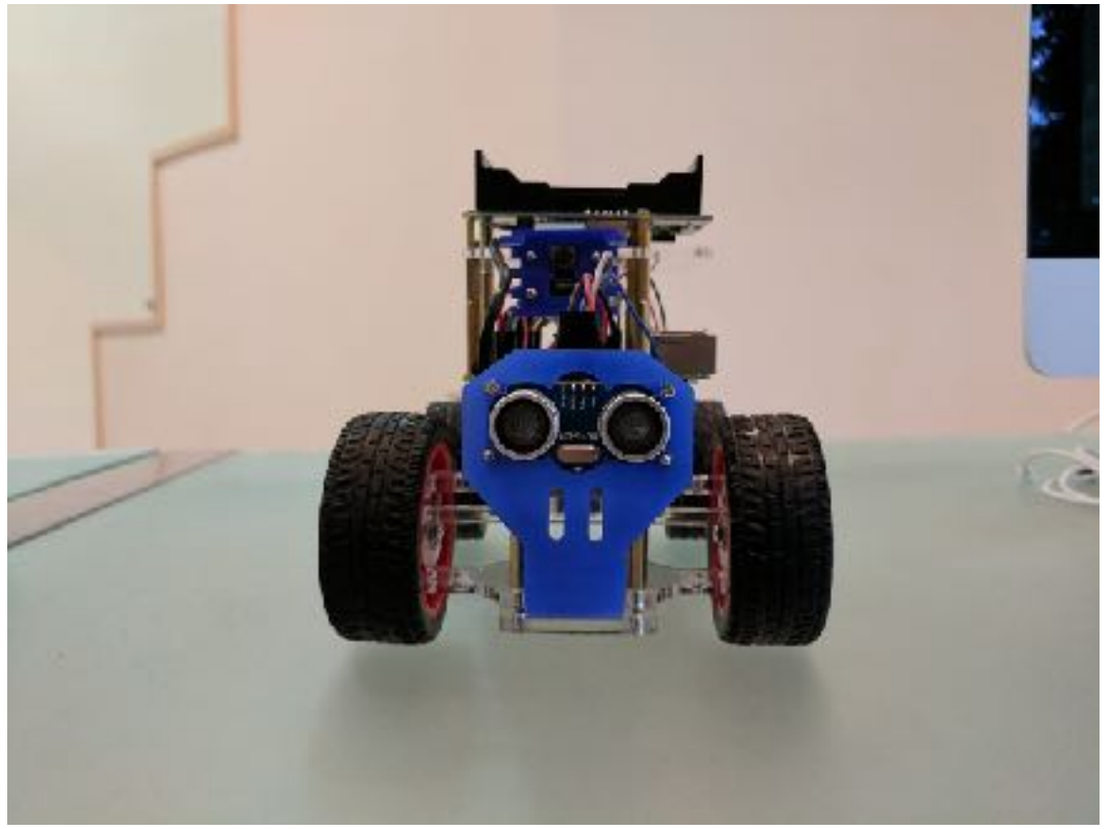
Machine Learning on a Raspberry Pi

Linda MacPhee-Cobb

https://github.com/timestocome

Mobile Robots

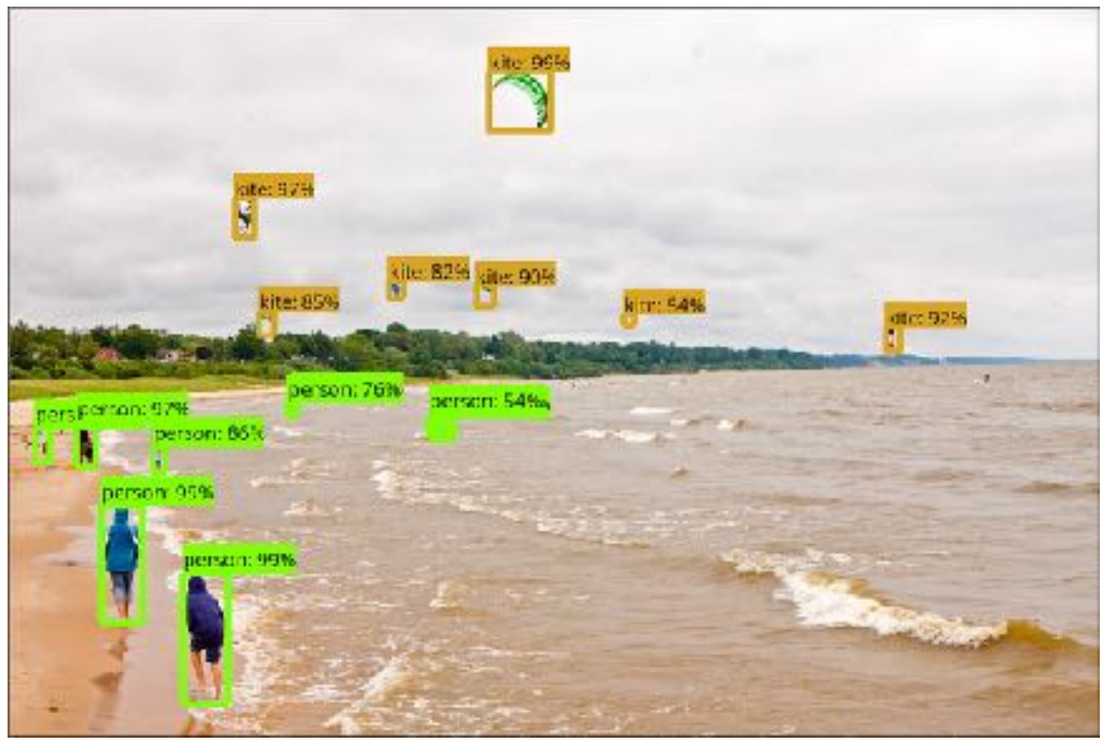




Sensors: UltraSonic distance sensor, Camera
Trained using Google's TF Mobile Models
Obstacle avoidance with simple RL

Output: Wheel movement f/r, turn servo to steer, LED lights if either cat is spotted

Image classification



Out of the box Google Mobile Network recognizes many objects ~ 10secs/frame
Re-trained on smaller sets (both cats) runs ~ 2 frames/second

~100 images each category, 3 categories (Min, Merlin, No cat)
~30 mins on desktop, copy trained network onto Pi

*** when you retrain it the built in classifications are forgotten**

Using Tensorflow on a Pi

Change the default system Python to 3.5

Follow the directions on
https://github.com/samjabrahams/tensorflow-on-raspberry-pi/

*** Change the name of the wheel file to cp35-cp35m from cp34-cp34**

Step by step detailed directions on
https://github.com/timestocome/RaspberryPi-Robot

Building an object detector

Download and install TensorFlow models
https://github.com/tensorflow/models

Install and Test the model
https://github.com/tensorflow/models/tree/master/research/object_detection

Take and image and try it out
https://github.com/timestocome/RaspberryPi-Robot/tree/master/ObjectDetection

Build a grey cat orange cat detector

You'll need Python and TF installed on your desktop

Collect images (~100 orange cat, ~100 grey cat, ~ 100 not a cat)

Put orange cat images, grey cat, not a cat images into 3 different directories

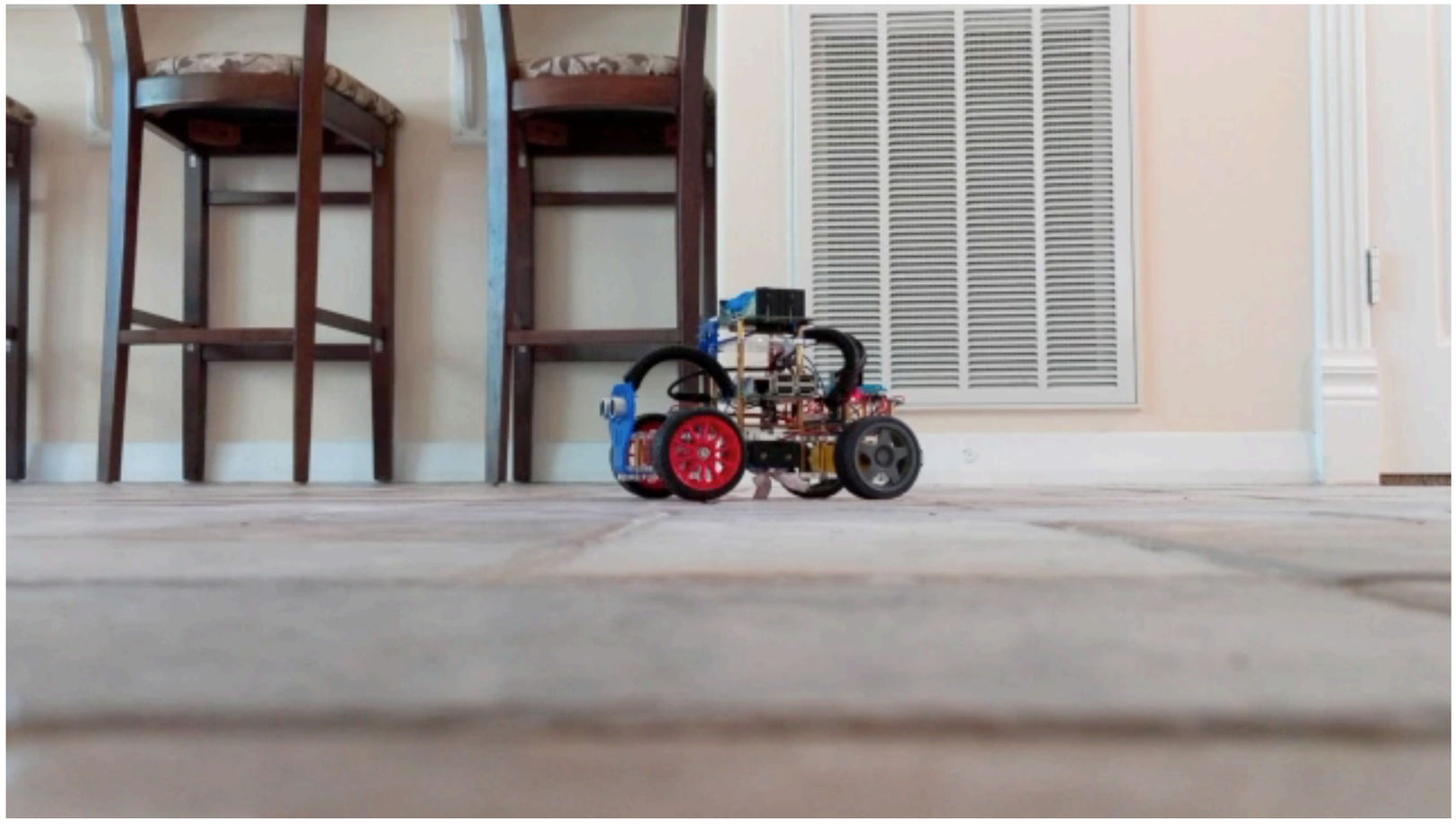
Try different models, you'l be trading speed and accuracy

copy the saved model to your Pi

Test

Detailed directions and help
https://github.com/timestocome/RaspberryPi-Robot/tree/master/catID

Self teaching



Using a small state space and one objective RL is easy
Multiple objectives is mostly unsolved

Simple Reinforcement Learning

Everything done on the Pi has to be lean.

It'll be taking sensor data at high speeds and in large amounts if you're using video

Start with random values

Every time the sensor data comes back into the range in your matrices select the action with the highest value. If things get better increase that value, else decrease it

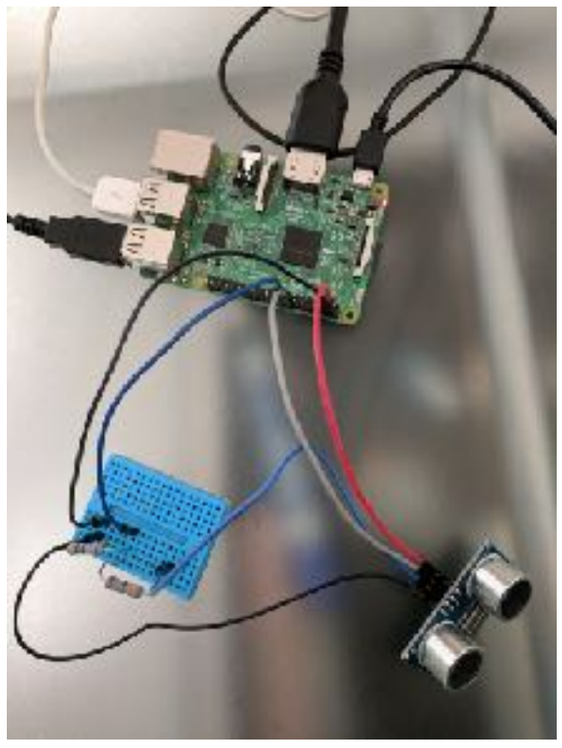
action/distance	1	2	3	4	5
forward	-0.9	0.7	0.4	-0.3	-0.4
right forward	0.7	0.8	-0.5	0.4	0.9
left forward	-0.2	-0.1	0.3	0.2	0.1
reverse	0.5	0.5	0.2	0.9	-0.6
right reverse	0.9	0.3	0.1	-0.5	0.4
left reverse	-0.6	0.1	-0.8	0.3	-0.2

UltraSonic Distance Sensor

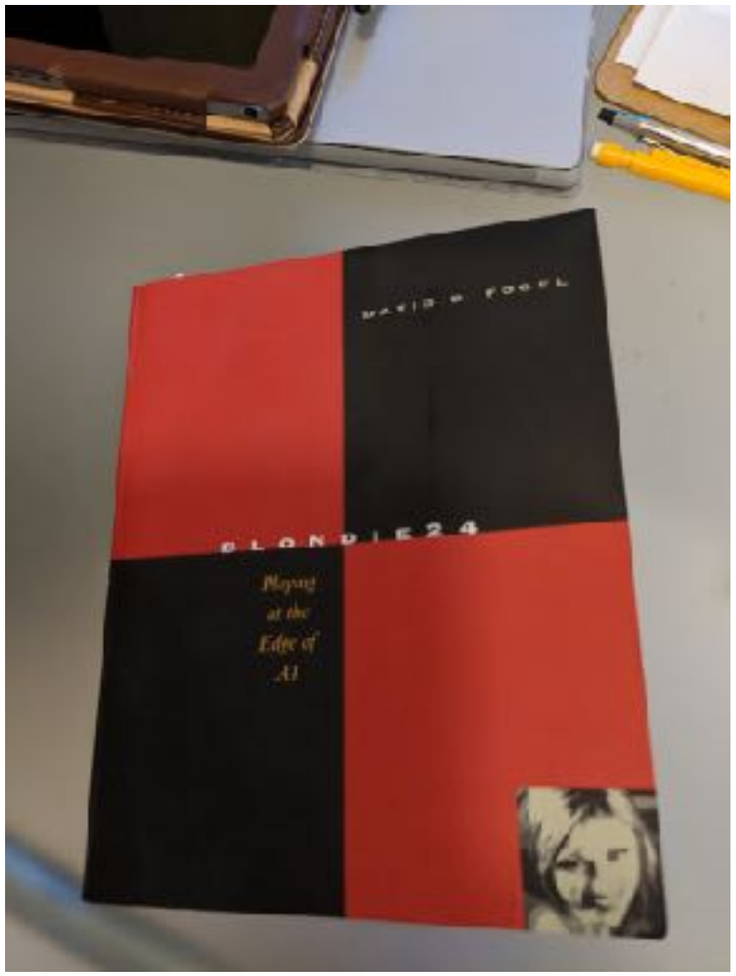
AI and ML Robotics Group (Stanford) trained several robots to map rooms using SLAM

Bumped into a RE Agent at EPO who was using these sensors to create floor plans by following walls

Car robot learns obstacle avoidance in about 100-1000 time steps, trick is to use a small state space matrix (actions, distance)



Artificial Intelligence



Before AlphaGo there was Blondie24

Everything AlphaGo does is in this book and a few things it doesn't yet do

Blondie24 played world class checkers on mid 1990s computers that were no more powerful than a Raspberry Pi

Tensorflow Resources

Tensorflow for Poets

https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/#0