MLiS II Project
Programming an Autonomous Driving Car

Dr Maggie Lieu, Dr Adam Moss

1 Overview

In the past decade, driver-less cars have gone from 'impossible' to 'inevitable'. Google's self driving car project[1] started in 2009 and now in 2021, most car manufacturers have adopted or are developing autonomous driving capabilities. The central concept behind the technology is to use trained object detection models to make principled decisions about what manoeuvres a car should make.

In this project your task is to train a deep learning algorithm to autonomously navigate a **real car** around a realistic test circuit, and make the appropriate manoeuvres where necessary. At the end of the project, you are expected to give a presentation and write a report about what you have done. Your model will be tested on the track and will compete against the models of your peers.

2 Challenge Details

In **pairs** you will develop a deep learning model that takes an image from a camera on the car, and predicts the appropriate speed and steering angle.

2.1 Training data

Due to Covid-19, we have generated training data for you to use. This is hosted on Kaggle (see next section) and consists of 13.8k images, along with the target car response (speed and steering angle).

2.2 Challenges

There will be two challenges:

1. A Kaggle hosted challenge. This will allow you to automate the process of model submission, and obtain an indication of performance (using a small set of test data), before we evaluate them on the final, unseen data. The Kaggle competition is hosted at `https://www.kaggle.com/t/464a3b674e20488b99181fdf5ec8fa90`. You are allowed 1 submission per day.

2. A live challenge, where your pre-trained model will be deployed to the car and tested on real circuits. This event will be live streamed so you can see what happens.

[1] `https://waymo.com/`

2.3 Equipment

The main body of the car is the SunFounder PiCar-V kit V2 (`https://www.sunfounder.com/smart-video-car-kit-v2-0.html`), and is equipped with a Raspberry Pi (RPi) 4. **Tensorflow v2.4** is installed on the car, so please make sure your model is capable with this version.

The car has an **optional** Coral Edge TPU, which is a custom device to run forward-pass operations for edge computing. The machine learning runtime used to execute models on the TPU is TensorFlow Lite v2.4. If you wish to use this capabiity, you will need to convert your code to TensorFlow Lite using the conversion tool `https://www.tensorflow.org/lite/convert`.

Note that it isn't necessary to convert your model to TensorFlow Lite. In this case it will run using regular TensorFlow on the RPi CPU, so you should bear in mind the CPU capability when performing inference on large image sizes. Furthermore, not all operations are compatible with Tensorflow Lite. Please see the link in the resources below for more details.

2.4 Deploying Models to the Car

A standardised skeleton code will be provided to you that you should integrate your pre-trained model with, which we will then install on the car prior to the live testing.

2.5 Training Machines

The machine allocations will be listed here soon.

As well as Google Colab and any local resources you may have, each group will have access to either the MLiS1 or MLiS2 machines (each with two 2080 Ti GPUs) to perform training. These are accessible by ssh'ing into the machine, by typing `ssh username@mlis1.nottingham.ac.uk` or `ssh username@mlis2.nottingham.ac.uk`, where `username` is your University username.

In order to install custom packages on your machine, you will need to set up a conda environment. To install conda, type the following command,

```
bash /shared/Anaconda3-2019.10-Linux-x86_64.sh
```

Once installed, you will need to add a start up script,

```
echo .  ~/.bashrc >> .profile
```

Lastly to create your conda environment use,

```
conda create --name my_env python=3.6
```

where `my_env` is the name of the environment. For more information on the usage of conda see: `https://docs.conda.io/projects/conda/en/latest/user-guide/cheatsheet.html`

2.6 Tracks

The training data was obtained on 3 tracks, as shown in Fig. 1. The live testing will also be performed on the same tracks.

2.7 Driving Scenarios

Important: we only use UK driving rules, i.e. driving on the left-hand side. The training data was based on the following driving scenarios:

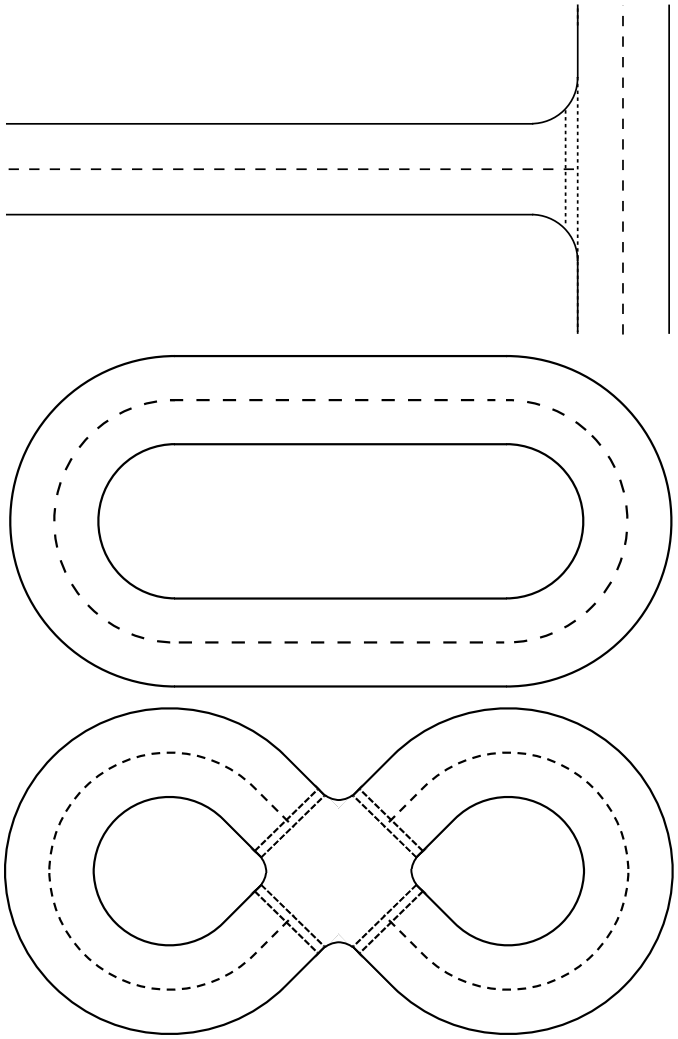

Figure 1: (Top) T-junction tracks (Middle) Oval Track (Bottom) Figure-of-eight track.

1. Keeping in lane driving along the straight section of the T-junction track.
2. As (1), but stopping if a pedestrian is in the road.
3. As (1), but driving (slowing down is optional) appropriately if pedestrians or other objects are on the side of (but not in) the road.
4. Driving around the oval track in both directions.
5. As (4), but stopping if a pedestrian is in the road.
6. As (4), but driving (slowing down is optional) appropriately if pedestrians or other objects are on the side of (but not in) the road.
7. Performing a turn at the T-junction, in response to a traffic sign (either left or right).
8. Driving around the figure-of-eight track in both directions, continuing straight at the intersection. We will not consider objects in or at the side of the road for this scenario.

9. Stopping at a red traffic light and continuing at a green traffic light.

We will **only** consider these scenarios in the live testing.

2.8 Prizes

The best performing car on the live testing will win a £100 Amazon voucher for the group.

3 Assessment and Important Dates

There are several important dates:

Kaggle and code submission The Kaggle deadline is **1.30pm on Thursday 6th May 2021**. You should also submit your model, fully implemented with the skeleton code to interface with the car, at the same time (the submission mechanism for the code is TBC).

Live testing This will take place on **Monday 10th May 2021** and will be live streamed. The Kaggle and live testing performance will contribute **20%** towards your total mark.

Presentations These will take place on **13th and 14th May 2021**. They will be performed in pairs via Microsoft Teams. The presentations will last for 25 minutes, during which you will be expected to speak for approximately 10 minutes each, with 5 minutes for questions. You will be penalized if you overrun significantly. The presentation will contribute **40%** towards your total mark and the assessment criteria are available on Moodle.

Report Your completed report should be submitted via Moodle by noon on **Friday May 21st 2021**. This is to be completed individually and will contribute **40%** towards your total mark. The report assessment criteria are available on Moodle.

4 Teams

Please come up with a suitable team name!

1. Patterson & Hall
2. Changlor & Lipisuntorn
3. Thrivikraman & Davy-Cripwell
4. Awasthi & Mirhosseini
5. Luo & Newton
6. Kim & Gonzalez
7. Naderi Nasab & Pacques

8. Roberts & Wilson
9. Mahmood & Lines
10. Wiech & Marshall-Hawkes
11. Nikolaidis & Black
12. Kenning
13. Zirereza & Rana
14. Ball & Li
15. Wilkinson & Georgiou
16. Bacon & Pache
17. Primrose & Boulaabi
18. Arif & Rashid

5 Resources

You may find the following resources helpful in designing your own model,

- `https://towardsdatascience.com/deeppicar-part-1-102e03c83f2c`
- `https://coral.ai/docs/edgetpu/models-intro/`

6 Tips

1. Make good use of your Kaggle submissions to test different things. You get 1 per day – don't let them go to waste! The best performing groups last year all made many submissions.
2. Get started early and make a plan.
3. Some initial data exploration on the training data will help you understand it better.

Acknowledgements

Special thanks to Simon Dye for assembling the cars, modifying the camera position and 3D printing the objects. Without him this project wouldn't be possible.