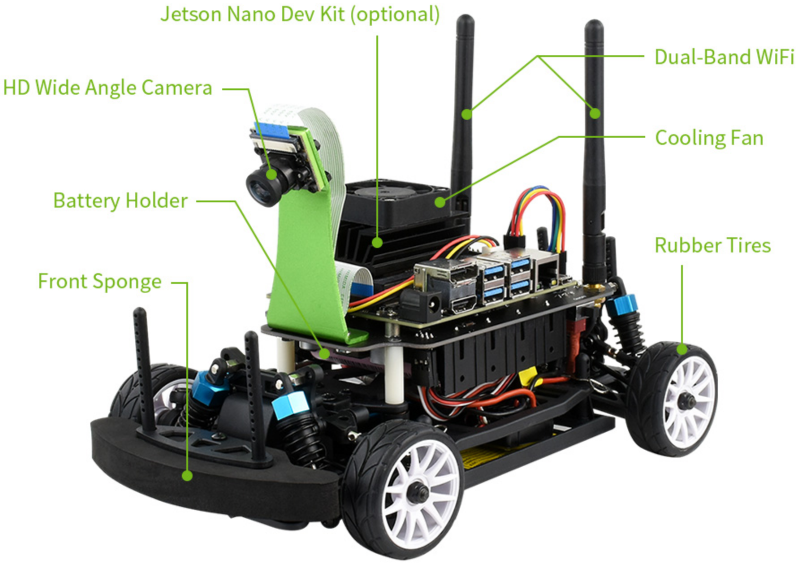


COMP47590

Advanced Machine Learning

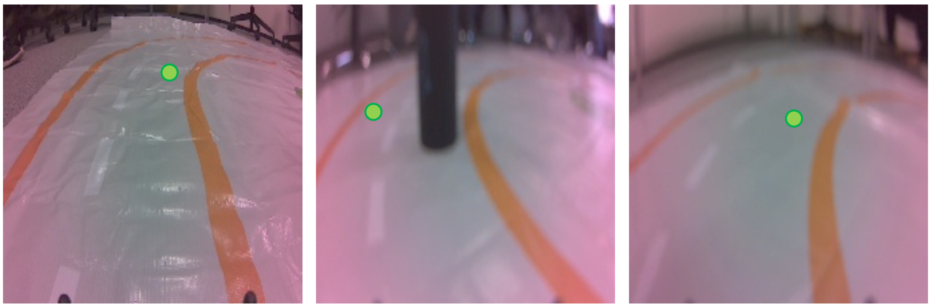
Assignment 1: Self Driving Racing Cars

Introduction

Autonomously driven vehicles have been long-promised[1], but are finally becoming a reality. This assignment will train a model to control a scale model self-driving car (a JetRacer Pro[2]) to drive around a defined racetrack.

Data

A dataset is provided to take a supervised learning approach to this problem. The dataset available for development is available to download here[3]. The dataset is composed of a collection of image files showing the view from the car's onboard camera. These images have been annotated with an optimal driving route. The annotations are in the form of an (X, Y) coordinate indicating the optimal racing line for a given scenario. Some sample images with annotations are shown below.



The annotations are contained in the image file names. A utility class (**XYDataset** contained in **xy_dataset.py**) is provided to help with loading this data and a starter notebook (**assignment_1_roboracer_ starter.ipynb**) demonstrating how to use this is also provided. This demo notebook creates training and validation partitions from the available data.

Tasks

Perform the following tasks:

1. Exam a selection of sample images from the training and validation datasets. sample images.
2. Train a **simple convolutional neural network** to predict the X and Y coordinates for the racing line based on an input image, and evaluate the performance of this model based on the validation data partition using an appropriate performance measure. For this , and all other, model runs use Weights & Biases logging to record model performance.
3. Using **transfer learning** based on a pre-trained CNN model (available through the **torchvision** package) build a more sophisticated CNN and evaluate the performance of this model based on the validation data partition using an appropriate performance measure.
4. Add **data augmentation** to the best model created in steps 2 and 3 to and evaluate the performance of this new model. Select a set of appropriate data augmentations to use.

[1] https://computerhistory.org/blog/where-to-a-history-of-autonomous-vehicles/

[2] https://www.waveshare.com/product/robotics/mobile-robots/jetson-nano-ai-robots/jetracer-pro-ai-kit.htm

[3] https://drive.google.com/drive/folders/1-0bXXkILLkrScLYmjITOfeS4Ib9mZB3j?usp=drive_link

5. **Evaluate** the performance of the best module built in steps 2, 3, and 4 on the test dataset (provided later in the assignment process).
6. Write a short (no more than 400 words) reflection on the performance of the different models built as part of this assignment. This reflection should take the form of a Weights & Biases report that summarises model performance and includes your reflection.

Notes

The following notes may be useful:

- **Can I Use Scikit-Learn, PyTorch, TorchVision, and Other Python Packages?** Yes – use whatever you think is useful.
- **Can I Work In A Team?** Teams of up to two people are allowed. All team members will receive the same mark. There is no penalty for submitting as a team, and no reward for submitting as an individual.
- **What Hardware Should I Use?** None of the datasets or models to be used in this assignment are so big that they shouldn't work fine on your laptop. If you are having problems, however, you should consider using Google Colab[4], a great online platform for machine learning.
- **Plagiarism.** Without looking too hard online you can find plenty of Gradient Boosting implementations that you could copy and submit for this assignment. **Don't do this!** Your submission should be your (or your team's (see below)) own work. If you are unsure about anything in this regard just ask.

[4] Google Colab https://colab.research.google.com

Submission

The key submission details for the assignment are as follows:

- **Submission date:** Friday 29th March 2024 before 23:30.
- **Submission method:** Submissions should be made through the module Brightspace site.
- **Submission format:** Submissions should compose a zip file containing the following:
  - a completed version of the template Jupyter notebook provided in .ipynb format (the Jupyter notebook and code should include adequate comments or markdown commentary to understand your implementation);
  - a HTML export of your Jupyter notebook after execution that contains all output;
  - a HTML export of your Weights & Biases reflection report;
  - both the Jupyter notebook and HTML files should be named as follows: studentID_fullName_assignment1
  - any other files required to execute your code.
- **Group submission:** For group submissions only one member of the group should submit.
- **Late submissions:** Late submissions will be penalised at 5% penalty per day.

Marking

Marking of tasks will be based on the following weighting.

Task 1	5%
Task 2	22%
Task 3	30%
Task 4	20%
Task 5	15%
Task 6	8%