Creating Robust Neural Network Generalisation for a Single-Sensor Self-Driving Car

Chara Grant

EERI 474 - Specification Document

Supervisor:	Prof Kenny Uren
Date:	April 22, 2020
Student number:	29458447

1 Document Information

Table 1: Document Information

Project Title	Creating Robust Neural Network Generalisation for a Single-Sensor Self-Driving Car
Document Number	CPDonSysReqV1.0
System	Convolutional Neural Network - based autonomous pilot and image preprocessing system
Document Issue Date	17-04-2020
Client	Caspar Lessing (Director) Counterpoint Dynamics
Client Reference	Project Littlefoot

Table 2: Origination and Approval

Checked By Party	Individual Name	Signature	Date Signed
Author:	Chara Grant		
Quality Assurance:	Prof. Kenneth Uren		
Client Approval:	Caspar Lessing/Alan Viljoen		
Technical Approval:	Prof. Kenneth Uren		
Technical Lead:	Alan Viljoen		
Project Manager:	Prof Leenta Grobler		

Table 3: Acceptance

Checked By Party	Individual Name	Signature	Date Signed
Client Approval:	Caspar Lessing/Alan Viljoen		
Technical Approval:	Prof. Kenneth Uren		
Technical Lead:	Alan Viljoen		

Table 4: Distribution List

Company	Individual Name	Date Distributed
Counterpoint Dynamics	Caspar Lessing	
Counterpoint Dynamics	Alan Viljoen	22-04-2020
Counterpoint Dynamics	Siyavuya Dambuza	
Counterpoint Dynamics	Chara Grant	17-04-2020
North-West University	Prof. Kenneth Uren	22-04-2020
North-West University	Prof. Pieter van Vuuren	
North-West University	F.R. Bezuidenhout	
North-West University	Prof. Leenta Grobler	22-04-2020

Table 5: Security Levels and Restrictions

Level	Description	Applicable Level
1	Strictly Confidential: Not to be distributed	
2	Company Confidential: Distributed inside company	
3	Client Confidential: Distributed to limited clients and contractors	x
4	Public Domain: Freely Distributed	

Table 6: Contact Information

Contact Person	Chara Grant
Company	Counterpoint Dynamics
Street Address	22 Reitz Street, Potchefstroom
Contact Number	079 314 4457
Email Address	chara.grant@cpdynamics.co.za
Company Web Site	www.cpdynamics.co.za

Table 7: Document Revision History

Responsible Person	**Description**	**Revision Number**
Chara Grant	Document Creation	0.0
Chara Grant	Added Glossary of Terms and Abbreviation List	0.1
Chara Grant	Main Client Requirements, Client Constraints and Levels of Compliance	0.2
Chara Grant	Detailed Client Requirements, Client Constraints and Levels of Compliance	0.3
Chara Grant	Measures of Success	0.4
Chara Grant	First Draft Released	1.0

Glossary

Artificial Neural Network A computational lattice of 'Linear Algebraic nodes', each composed of data matrices, weight and bias matrices, and non-linearization functions..

Client Refers to the individual/company for which the system will be developed. In this context the Client will refer to Counter Point Dynamics and their relevant members.

Donkey Car Ecosystem Refers to all aspects of the Donkey Car Library itself - including scripts, tools, structure, and programming language. The Donkey Car Ecosystem makes use of the Donkey Car Standard (not named such by original developers, the term Donkey Car Standard is used by the author of this document) available throughout the Donkey Car Docs page, especially under the Parts heading of the page..

Drive Loop Main control sequence of the Donkey Car Software - includes timing, calling of relevant functions, ensuring correct order of processing is followed, and controls the flow of data between classes, functions and components..

MIT Open Source License An Open Source license specifying that the software licensed under it may be freely used, replicated, and sold provided that original credit is given to the creators and a disclaimer is placed in the source code..

Model Derived in understanding from the term referring to a Statistical Model. In this context, Model refers to an Artificial Neural Network Architecture and its weights and biases trained to perform certain computations.

Acronyms

ANN Artificial Neural Network.

CC Client Constraint.

CL Compliance Level.

CNN Convolutional Neural Network.

CPD Counterpoint Dynamics.

CR Client Requirement.

IMU Inertial Measurement Unit.

MOS Measure of Success.

PWM Pulse Width Modulation.

px Pixels.

Links of Interest

DIYRobocars Movement https://diyrobocars.com/.
Donkey Car Docs Page https://docs.donkeycar.com/.
Donkey Car GitHub repository https://github.com/autorope/donkeycar.
Donkey Car Standard https://docs.donkeycar.com/parts/about/.

Open Source Definition https://opensource.org/osd-annotated.

The MIT License https://opensource.org/licenses/MIT.

Table of Contents

1 Document Information **i**

2 Introduction **1**

3 Problem Statement **1**

4 Client **2**

4.1 Client Intentions 2
4.2 Client Requirements 2
4.3 Client Constraints 3
4.4 Client Compliance Levels 3

5 Requirements **4**

5.1 CR 1: An End-To-End Self-Driving Pilot must be developed 4
5.1.1 CR 1.1: Pilot Shall be Capable of At Least Autonomous Turning . . . 4
5.1.2 CR 1.2: Pilot Shall Conform to Donkey Car System Standard 4
5.2 CR 2: Each track and track shape used shall use the same set of standard markings 4
5.2.1 CR 2.1: Each track shape shall conform to a DIYRobocars-compliant standard 4
5.2.2 CR 2.2: The set of standard road markings to be used shall be found in use on South African roads 4
5.2.3 CR 2.3: Edges and boundaries of track and driving space shall be clearly identifiable 5
5.3 CR 3: Methods shall be introduced that improve the visual lane-detection capabilities of the Pilot over the standard 5
5.3.1 CR 3.1: Methods may include edge detection, image thresholding, semantic segmentation or similar solutions. 5
5.3.2 CR 3.2: Lane detection methods employed shall be performed on the raw input images. 5
5.3.3 CR 3.3: Lane detection shall be shown to improve the performance of a model over a model fed with raw data in a controlled experiment. . . 5
5.4 CR 4: No adverse conditions shall be allowed in the training, testing, and demonstration of the Pilot 5
5.4.1 CR 4.1: The Pilot shall have full view of the track at all times 5
5.4.2 4.2: Besides aiding in the robustness of the model - adverse examples shall not be specifically required in training 5

5.4.3 4.3: Besides where requested in testing by the developer, obstacles and confounding entities may not be put in/near the vehicle's path 5
5.5 CR 5: Software shall be licensed under the MIT Open Source License, and CPD Shall have access to information generated 5
5.5.1 CR 5.1: All software developed for and as the Pilot shall be licensed under the MIT Open Source License 5
5.5.2 CR 5.2: CPD and its members shall have free access to read, derive from, and distribute documentation and written works generated in this project 6
5.6 CR 6: The Pilot shall be based on an Artificial Neural Network Machine Learning system 6
5.6.1 CR 6.1: The Artificial Neural Network Architecture may be composed of a single stage or a multi-stage architecture 6
5.6.2 CR 6.2: The ANN Architecture used must be the main decision-making driver of the system 6
5.6.3 CR 6.3: The ANN Architecture may be trained via either Online Learning or Offline learning at developer's discretion 6
5.7 CR 7: The Pilot shall be developed to be 'Plug and Play' Compatible with the Donkey Car system 6
5.7.1 CR 7.1: See CR 1.2.4 7
5.7.2 CR 7.2: Development of the Pilot shall take place exclusively in the Python Programming Language 7
5.8 CR 8: The Donkey Car system and its associated simulators shall be employed as the testing platform of the Pilot 7
5.8.1 CR 8.1: The Pilot may be developed independently from the Donkey Car System in tools such as Jupyter Notebook 7
5.8.2 CR 8.2: The Pilot may be unit-tested independently from the Donkey Car System 7
5.8.3 CR 8.3: The final Integration and Testing of the Pilot shall conform to the stipulations set out in CR 7 7
5.9 CR 9: The sensor to be used shall be the already-included front-facing camera 8
5.9.1 CR 9.1: The current camera is a Raspberry Pi Fisheye lens camera. If compatibility issues with the Jetson Nano board arise, or if a better option for the front facing camera presents itself, this camera may be replaced. 8
5.10 CR 10: Additional Sensors may be used as long as they do not require significant changes to the Donkey Car System and surrounding environment, and as long as the scope of the project allows 8
5.10.1 CR 10.1: Additional sensors used shall not detract from the Goals of the project 8

6 Constraints **8**
6.1 CC 1: At least one Neural Network shall be present in the solution and Machine Learning Concepts shall be used . . . 8
6.1.1 CC 1.1: Neural Network shall refer to a complete ANN Architecture with a clearly defined input, output and purpose . . . 9
6.1.2 CC 1.2: Machine Learning concepts shall include, but are not limited to: 9
6.2 CC 2: Donkey Car is the testing platform for any Pilot developed. . . . 9
6.3 CC 3: Industry standard tools must be used. The programming language of choice is Python . . . 9
6.3.1 CC 3.1: Machine-Learning tools that shall be used include but are not limited to: . . . 9
6.3.2 CC 3.2: At least Python 3.6 shall be used . . . 9
6.3.3 CC 3.3: Environments and tools such as Jupyter Notebook, Anaconda, and neptune.ai may be used to improve productivity and allow tracking of design and experimentation . . . 9
6.4 CC 4: Hardware used shall be compatible with the Donkey Car System . . . 9
6.4.1 CC 4.1: The Donkey Car Documentation lists compatible boards to be the Raspberry Pi (Pi 3B+, Pi 4) and the Jetson Nano . . . 9
6.4.2 CC 4.2: Any Cameras, Servo HATs, or additional sensors used shall be compatible with either a Raspberry Pi or a Jetson Nano board . . . 9

7 Compliance Levels and Measures of Success **10**
7.1 CL 1: The Pilot is able to steer the vehicle around a single track in different lighting conditions . . . 10
7.2 CL 2: The Pilot is able to steer the vehicle around a set of different tracks in different lighting conditions . . . 10
7.3 CL 3: The Pilot is able to steer the vehicle around a set of different tracks in different lighting conditions and different environments . . . 11
7.4 Measures of Success . . . 12

List of Figures

List of Tables

1 Document Information i
2 Origination and Approval i
3 Acceptance i
4 Distribution List ii
5 Security Levels and Restrictions ii
6 Contact Information ii
7 Document Revision History iii

2 Introduction

This document details the requirements and specifications pertaining to the design of an autonomous self-driving car pilot and image preprocessing system, henceforth referred to as the Pilot. This project has been initiated by Counterpoint Dynamics and is taken on with assistance from resources at the NWU.

The Pilot in question will be developed to be 'Plug-and-Play' compatible with the Donkey Car system. The Donkey Car system is a set of Python-based machine learning libraries and utilities, designed to operate on a Raspberry Pi or Jetson Nano. With a Servo HAT attached, this system is able to provide control to an electric, scale, Remote-Controlled hobby car. With a Neural Network present in the control loop of the system the car is able to be autonomously piloted. This system is entirely open-source and is licensed under the MIT open source license. The documentation of this open-source project, as well as the GitHub repository, are found in the links table at the beginning of this document.

The current Donkey Car platform revolves around a Neural Network autonomous pilot - able to recognize certain features in images and produce outputs based on these recognized features. These outputs are used to steer and propel the vehicle autonomously. The implementation of the Donkey Car System on which this project is based makes use of Behavioural Cloning to produce a trained Neural Network model. This method consists of manually driving the car around a track for several laps while thousands of images and driving angle records are collected. Once enough images and records are collected, these are used to train the vehicle to follow the lanes in the same way that the human driver had.

This current implementation is only able to drive in environments and on track shapes that it was trained on. Any deviation in any detail of the environment causes the pilot to behave erratically and veer off the track. This implementation also requires many thousands (15000+) in order to properly be trained - implying a very low practicability in real-world situations where data may or may not be limited.

3 Problem Statement

The main goal of this project is to create a Pilot (Neural Network and Image Preprocessing System) that will allow a Donkey Car to **reliably** follow **a** track. A track that is followed needs to form part of a set of tracks that do not have the same shape. This performance must also be replicated in varied environments - with different lighting conditions. Simply put - this Pilot must be able to recognize lane lines and feed corresponding outputs to the rest of the Donkey Car system in order to produce reliable driving. This is not a performance orientated problem - meaning that the speed and/or elegance with which the Donkey Car follows the track is of no consequence. This is an exercise in producing robust Neural Network Generalization in the context of piloting a single-sensor hobby vehicle around a standard-marking track in varying environments and conditions.

4 Client

The Client for which the Pilot is being developed is Counterpoint Dynamics. Counterpoint Dynamics is a Consultancy Firm based in Johannesburg, South Africa. They specialise in Quantative Analysis, Machine Learning, Market and Credit risk, as well as other disciplines related to the Financial Sector.

4.1 Client Intentions

The scope of this project tends more to research and technical enrichment than to product development. The Pilot that is to be developed is viewed as a research case study for Neural Network Generalisation - and the results of this study will be used to further the practicability and reach of future projects that Counterpoint Dynamics will take on.

4.2 Client Requirements

The Client's requirements for this project are specified as follows:

CR 1: An end-to-end Pilot, as defined by other working examples within the Donkey Car framework, must be developed. This Pilot must be able to navigate a car around a track.

CR 2: Each track and track shape used must have the same set of standard markings. No special/unique markings not readily found in real-world road settings can be used. Typically, white and yellow lines along with common road signs and markings such as arrows may be used.

CR 3: Improvements shall be made on the visual lane-detection on incoming image data by introducing methods such as edge detection, image thresholding, semantic segmentation or similar solutions.

CR 4: No adverse conditions. The vehicle must have full view of the portion of track in front of it at all times. No obstacles or confounding entities of any kind are permitted to be placed in front of/around the car unless specifically called for in the test conducted by developer.

CR 5: All software generated in this project must be licensed under the MIT open-source license, and Counterpoint Dynamics must have access to any documentation and informational content generated.

CR 6: The Pilot must be based on an Artificial Neural Network Machine Learning system - since this is research case specifically into the field of Neural Networks and Machine Learning.

CR 7: The Pilot shall be developed to be plug-and-play compatible with the Donkey Car system.

CR 8: The Donkey Car system, as well as its associated simulators, shall be employed as the testing platform for the Pilot

CR 9: The sensor to be used shall be the already-included single front fisheye-lens raspberry Pi Camera.

CR 10: Additional sensors (such as an IMU sensor) that do not require that changes be made to the Donkey Car system or to the physical environment surrounding car may be used as scope and time constraints allow. These sensors may not detract from or interfere with the Neural Network Generalization exercise.

4.3 Client Constraints

The constraints placed by the Client on the project are defined as follows:

CC 1: At least one Neural Network shall be present in the solution and Machine Learning concepts shall be used.

CC 2: Donkey Car is the testing platform for any Pilot developed. Consequently, any Pilot developed must be compatible with the Donkey Car system and function seamlessly within it.

CC 3: Industry standard tools - such as Tensor Flow, Keras and similar must be used (to be further specified). The programming language of choice is Python.

CC 4: Hardware used must be compatible with the Donkey Car system (Raspberry Pi or Jetson Nano compatible).

If an alternative to the industry standard tools is found, and there is sufficient time for reskilling the developer and porting solutions to the new tools, then alternative tools may be used. However, it is strongly advised that focus is placed on the functionality and core objective of the project. Alternative tools, 'methods of attack', algorithms etc. may be listed and documented as topics for later/further research.

4.4 Client Compliance Levels

It has been specified by the client that compliance to the Project Goal and requirements be measured in stages, or levels. Each level describes an adherence to certain facets of the solution space. The suggested levels of compliance were suggested to follow the pattern:

CL 1: The Pilot is able to steer the vehicle around a single track (same shape, markings, length) in different lighting conditions. (sunrise, morning, noon, afternoon, dusk etc.). The track is in precisely the same location at each test run.

CL 2: The Pilot is able to steer the vehicle around a fixed set of differing tracks (same markings, different lengths, different shapes) in different lighting conditions. (sunrise, morning, noon, afternoon, dusk etc.) The track is in precisely the same location at each test run.

CL 3: The Pilot is able to steer the vehicle around a fixed set of differing tracks in different lighting conditions. Test runs must now take place in at least three different locations/environments (indoor and outdoor locations are included).

CL 4: Pilot is able to steer according to requirements in CL 3. It is also able to recognize and react to a limited set of road signs.

CL 5: Pilot is able to follow road markings that are not assigned to a fixed track. I.e., it must be able to follow the road markings used on training tracks that are now laid out on campus pathways, hallway floors, lecture hall floors etc. in a randomized route.

CL 6: Pilot is able to follow randomized route described in CL 5. In addition, it is able to recognize and react to a predefined set of standard road signs.

These levels can be expanded on and added to. The Client suggested that for the scope of this development of this project that the level of compliance be evaluated until Level 3. The remaining levels can be used as guidelines for where to take the project in the future.

5 Requirements

5.1 CR 1: An End-To-End Self-Driving Pilot must be developed

5.1.1 CR 1.1: Pilot Shall be Capable of At Least Autonomous Turning

5.1.2 CR 1.2: Pilot Shall Conform to Donkey Car System Standard

CR 1.2.1: Pilot Shall Include Image Processing

CR 1.2.2: Pilot Shall Provide Angle and Throttle Values as Output

CR 1.2.3: Pilot Shall Contain at Least one Neural Network Architecture which governs the piloting process

CR 1.2.4: Pilot Shall Conform to Structure of Donkey Car Ecosystem

CR 1.2.4.1 Neural Network Architecture Shall be Defined as per Convention found in Keras.py file

CR 1.2.4.2 Neural Network Architecture Shall be stored and referenced according to Standard of other Donkey Car Pilot Models

5.2 CR 2: Each track and track shape used shall use the same set of standard markings

5.2.1 CR 2.1: Each track shape shall conform to a DIYRobocars-compliant standard

5.2.2 CR 2.2: The set of standard road markings to be used shall be found in use on South African roads

CR 2.2.1: Requirement pertaining to road-surface markings

CR 2.2.2: Requirement pertaining to road-sign markings

5.2.3 CR 2.3: Edges and boundaries of track and driving space shall be clearly identifiable

5.3 CR 3: Methods shall be introduced that improve the visual lane-detection capabilities of the Pilot over the standard

5.3.1 CR 3.1: Methods may include edge detection, image thresholding, semantic segmentation or similar solutions.

5.3.2 CR 3.2: Lane detection methods employed shall be performed on the raw input images.

5.3.3 CR 3.3: Lane detection shall be shown to improve the performance of a model over a model fed with raw data in a controlled experiment.

5.4 CR 4: No adverse conditions shall be allowed in the training, testing, and demonstration of the Pilot

5.4.1 CR 4.1: The Pilot shall have full view of the track at all times

CR 4.1.1: The recognizability of any markings used shall not be purposely obscured

CR 4.1.2: The lighting in the environment shall not obscure any vital features

5.4.2 4.2: Besides aiding in the robustness of the model - adverse examples shall not be specifically required in training

5.4.3 4.3: Besides where requested in testing by the developer, obstacles and confounding entities may not be put in/near the vehicle's path

5.5 CR 5: Software shall be licensed under the MIT Open Source License, and CPD Shall have access to information generated

5.5.1 CR 5.1: All software developed for and as the Pilot shall be licensed under the MIT Open Source License

CR 5.1.1: The GitHub repository housing the project code (the Pilot and all supporting scripts and software) shall be listed as licensed under the MIT Open Source license.

CR 5.1.2: The MIT License shall be used as specified at https://opensource.org/licenses/MIT

CR 5.1.3: The MIT License and its disclaimer shall appear at least in part in all design, descriptive, and reporting documentation generated on the Pilot

5.5.2 CR 5.2: CPD and its members shall have free access to read, derive from, and distribute documentation and written works generated in this project

5.6 CR 6: The Pilot shall be based on an Artificial Neural Network Machine Learning system

5.6.1 CR 6.1: The Artificial Neural Network Architecture may be composed of a single stage or a multi-stage architecture

5.6.2 CR 6.2: The ANN Architecture used must be the main decision-making driver of the system

5.6.3 CR 6.3: The ANN Architecture may be trained via either Online Learning or Offline learning at developer's discretion

The data used to train the model will be available in its entirety. Therefore - either option is viable, each with their own pros and cons.

CR 6.3.1: Data available to the ANN for training shall be image data and json records

CR 6.3.1.1: The images available shall be obtained from the Pi Camera and PiCamera class during the recording drive loop of the Donkey Car. Images available for use from other Donkey Car users or online data repositories may also be used.

CR 6.3.1.2: The images used shall be of shape 160 px by 120 px, as is the standard in previously developed Donkey Car Pilots.

CR 6.3.1.3: The json records shall contain the following fields: corresponding image filename, relative throttle value, relative angle value.

CR 6.3.1.4: The relative angle and throttle values are expressed as a fraction of the full range of PWM values used: they shall appear as floating point values. These floating point values shall appear in the range of -1.00 to 1.00.

CR 6.3.2: Training methods, optimization functions, and model 'hyperparameters' and their affect on the Pilot's performance and accuracy must be logged

5.7 CR 7: The Pilot shall be developed to be 'Plug and Play' Compatible with the Donkey Car system

5.7.1 CR 7.1: See CR 1.2.4

CR 7.1.1: Each discrete Pilot functionality as defined by the developer as such shall be developed in its own 'Part' class and included in the 'Vehicle' drive loop as per Donkey Car standard

CR 7.1.1.1: Each 'part' class newly set up, or added to, for each discrete pilot functionality shall conform to the Donkey Car Part Class convention.

CR 7.1.1.2: Interactions between 'part' classes created or used in the development shall be governed by Donkey Car conventions present and seen in current Donkey Car implementations.

CR 7.1.1.3: Creation of a new control script in which the instantiation of the 'Vehicle' class is present shall conform to Donkey Car conventions and shall ensure compatibility with the existing Drive Loop.

CR 7.1.2: Additions to or recreation of the Vehicle control script shall adhere to Donkey Car conventions and follow current Donkey Car implementations

5.7.2 CR 7.2: Development of the Pilot shall take place exclusively in the Python Programming Language

5.8 CR 8: The Donkey Car system and its associated simulators shall be employed as the testing platform of the Pilot

5.8.1 CR 8.1: The Pilot may be developed independently from the Donkey Car System in tools such as Jupyter Notebook

5.8.2 CR 8.2: The Pilot may be unit-tested independently from the Donkey Car System

5.8.3 CR 8.3: The final Integration and Testing of the Pilot shall conform to the stipulations set out in CR 7

CR 8.3.1: The Donkey Gym Simulation Environment shall be used for end-to-end testing of an Integrated Pilot

CR 8.3.1.1: The Donkey Gym Simulation Environment shall be used to test the Pilot prototypes developed before real-world testing is undertaken.

CR 8.3.1.2: The Donkey Gym Simulation Environment may be used for data collection.

CR 8.3.1.3: In case of extended COVID-19 Pandemic Lockdown and/or restrictions, the Donkey Gym Simulation Environment shall be used as a virtual and remote demonstration platform.

CR 8.3.2: Real-World testing and demonstration of an Integrated Pilot shall take place on a physical Donkey Car system

CR 8.3.2.1: The physical Donkey Car system in question shall consist of a Raspberry Pi (or Jetson Nano), a single front-facing camera, and a Servo HAT mounted securely to an electric remote-control hobby car. (Scale between 1/16 and 1/10 according to hobby car scaling definitions)

CR 8.3.2.2: The Donkey Car Software, complete with the Integrated Pilot, shall be installed on the Control Board (Raspberry Pi or Jetson Nano) of the Donkey Car System.

CR 8.3.2.3: (See CR 2 and CR 2.1) Tracks used for testing and demonstration shall adhere to DIYRobocars racing standards.

5.9 CR 9: The sensor to be used shall be the already-included front-facing camera

5.9.1 CR 9.1: The current camera is a Raspberry Pi Fisheye lens camera. If compatibility issues with the Jetson Nano board arise, or if a better option for the front facing camera presents itself, this camera may be replaced.

5.10 CR 10: Additional Sensors may be used as long as they do not require significant changes to the Donkey Car System and surrounding environment, and as long as the scope of the project allows

5.10.1 CR 10.1: Additional sensors used shall not detract from the Goals of the project

CR 10.1.1: Additional sensors shall not interfere with the Neural Network Generalization exercise: i.e. data collected from these sensors shall not interfere with the visual data on which the Neural Network should be able to generalize its predictions

CR 10.1.2: The use of additional sensors shall not take priority over or detract from the design and implementation of the Pilot when it comes to troubleshooting, integration, or development

6 Constraints

6.1 CC 1: At least one Neural Network shall be present in the solution and Machine Learning Concepts shall be used

See CR 1.2.3, CR 1.2.4, CR 4 and CR 6.

6.1.1 CC 1.1: Neural Network shall refer to a complete ANN Architecture with a clearly defined input, output and purpose

6.1.2 CC 1.2: Machine Learning concepts shall include, but are not limited to:

CR 1.2.1: The type of Machine Learning taking place shall be Supervised Learning, with appropriately labeled training, vaildation, and testing datasets.

CR 1.2.2: Models shall be trained using industry standard training workflows at the discretion of the developer.

CR 1.2.3: Each trained Model shall be evaluated according to its loss, accuracy, and other Machine Learning metrics.

6.2 CC 2: Donkey Car is the testing platform for any Pilot developed.

See CR 1.2 and CR 7.

6.3 CC 3: Industry standard tools must be used. The programming language of choice is Python

6.3.1 CC 3.1: Machine-Learning tools that shall be used include but are not limited to:

CC 3.1.1: Tensorflow 1.13.0, or any more recent version, provided it corresponds with the default version installed with the Donkey Car Software on-board the Raspberry Pi or Jetson Nano.

CC 3.1.2: The Keras API shall be used in conjunction with Tensorflow

CC 3.1.3: Numpy, Scipy, and other Python Libraries shall be used as becomes necessary

6.3.2 CC 3.2: At least Python 3.6 shall be used

6.3.3 CC 3.3: Environments and tools such as Jupyter Notebook, Anaconda, and neptune.ai may be used to improve productivity and allow tracking of design and experimentation

6.4 CC 4: Hardware used shall be compatible with the Donkey Car System

6.4.1 CC 4.1: The Donkey Car Documentation lists compatible boards to be the Raspberry Pi (Pi 3B+, Pi 4) and the Jetson Nano

6.4.2 CC 4.2: Any Cameras, Servo HATs, or additional sensors used shall be compatible with either a Raspberry Pi or a Jetson Nano board

7 Compliance Levels and Measures of Success

The compliance levels as applicable to the immediate development of a Pilot as defined in the Problem Statement (see Section 3) are CL 1 through CL 3. These are the compliance levels that will be detailed in this section.

7.1 CL 1: The Pilot is able to steer the vehicle around a single track in different lighting conditions

Requirements for compliance to CL1:

CL 1.1: A track of identical shape, dimensions and markings is used at each test.

CL 1.2: The Pilot is tested in outdoor and indoor lighting at different times of day:

CL 1.2.1: The Pilot is tested outdoors, at a single chosen location, with no shade cast on track:

CL 1.2.1.1: At early-morning light. (6:00 during summer, 7:00 during winter)

CL 1.2.1.2: At morning light. (10:00 to 11:00)

CL 1.2.1.3: At noon. (12:00)

CL 1.2.1.4: At early afternoon. (13:00 during winter, 14:00 in summer)

CL 1.2.1.5: At late afternoon. (15:00 during winter, 17:00 during summer)

CL 1.2.2: The Pilot is tested outdoors, at a single chosen location, with variable shade cast on track:

CL 1.2.2.1: At early-morning light. (6:00 during summer, 7:00 during winter)

CL 1.2.2.2: At morning light. (10:00 to 11:00)

CL 1.2.2.3: At noon. (12:00)

CL 1.2.2.4: At early afternoon. (13:00 during winter, 14:00 in summer)

CL 1.2.2.5: At late afternoon. (15:00 during winter, 17:00 during summer)

CL 1.2.3: The Pilot is tested indoors, in a single chosen sparsely-furnished room, with flourescent 'office' lighting.

7.2 CL 2: The Pilot is able to steer the vehicle around a set of different tracks in different lighting conditions

The Requirements for compliance to CL2:

CL 2.1 A single chosen environment is used for each test type (indoor, outdoor unshaded, outdoor shaded).

CL 2.2 At least three track shapes are used in each test.

CL 2.3 The Pilot is tested in outdoor and indoor lighting at different times of day:

CL 2.3.1: The Pilot is tested outdoors, at a single chosen location, on at least three different track shapes, with no shade cast on track:

CL 2.3.1.1: At early-morning light. (6:00 during summer, 7:00 during winter)

CL 2.3.1.2: At morning light. (10:00 to 11:00)

CL 2.3.1.3: At noon. (12:00)

CL 2.3.1.4: At early afternoon. (13:00 during winter, 14:00 in summer)

CL 2.3.1.5: At late afternoon. (15:00 during winter, 17:00 during summer)

CL 2.3.2: The Pilot is tested outdoors, at a single chosen location, on at least three different track shapes, with variable shade cast on track:

CL 2.3.2.1: At early-morning light. (6:00 during summer, 7:00 during winter)

CL 2.3.2.2: At morning light. (10:00 to 11:00)

CL 2.3.2.3: At noon. (12:00)

CL 2.3.2.4: At early afternoon. (13:00 during winter, 14:00 in summer)

CL 2.3.2.5: At late afternoon. (15:00 during winter, 17:00 during summer)

CL 2.3.3: The Pilot is tested indoors, in a single chosen sparsely-furnished room, on at least three different track shapes, with flourescent 'office' lighting

7.3 CL 3: The Pilot is able to steer the vehicle around a set of different tracks in different lighting conditions and different environments

The Requirements for compliance to CL2:

CL 3.1 At least three different chosen environments are used for each test type (indoor, outdoor unshaded, outdoor shaded).

CL 3.2 At least three track shapes are used in each test.

CL 3.3 The Pilot is tested in outdoor and indoor lighting at different times of day:

CL 3.3.1: The Pilot is tested outdoors, at a each chosen location, on at least three different track shapes, with no shade cast on track:

CL 3.3.1.1: At early-morning light. (6:00 during summer, 7:00 during winter)

CL 3.3.1.2: At morning light. (10:00 to 11:00)

CL 3.3.1.3: At noon. (12:00)

CL 3.3.1.4: At early afternoon. (13:00 during winter, 14:00 in summer)

CL 3.3.1.5: At late afternoon. (15:00 during winter, 17:00 during summer)

CL 3.3.2: The Pilot is tested outdoors, at a each chosen location, on at least three different track shapes, with variable shade cast on track:

CL 3.3.2.1: At early-morning light. (6:00 during summer, 7:00 during winter)

CL 3.3.2.2: At morning light. (10:00 to 11:00)

CL 3.3.2.3: At noon. (12:00)

CL 3.3.2.4: At early afternoon. (13:00 during winter, 14:00 in summer)

CL 3.3.2.5: At late afternoon. (15:00 during winter, 17:00 during summer)

CL 3.3.3: The Pilot is tested indoors, in each chosen sparsely-furnished room, on at least three different track shapes, with flourescent 'office' lighting

7.4 Measures of Success

As each of the tests are conducted, it is important to recognize what constitutes a successful trial. Some criteria with which to measure successful autnomous steering on the part of the Pilot are as follows:

MOS 1 The vehicle does not veer off track more than twice. (i.e., all four wheels do not leave the boundaries of the lane lines more than twice.)

MOS 2 In the event that the vehicle has veered off track, it is able to correct itself back within the boundaries of the lane lines at least once.

MOS 3 In the event that the vehicle's speed is also controlled by the Pilot, the Pilot should complete a lap in at least twice as long as an average human lap time.

MOS 4 In the event that the vehicle's speed is controlled by a human co-driver, the Pilot should complete a lap in at least one and a half times the average human lap time.

MOS 5 When the vehicle's path is plotted, the path that the Pilot takes should correspond at least 80% or more with the average human path.

Each final test conducted in pursuit of achieving each compliance level should be filmed and recorded. The data collected during each test should also be stored and organized. This is for the purposes of augmenting and simplifying the demonstration process on Project Day.