Self-Driving Car Engineer Syllabus

Contact Info

While going through the program, if you have questions about anything, you can reach us at support@udacity.com. For help from Udacity Mentors and your peers visit the Udacity Classroom.

Nanodegree Program Info

This program will teach you the techniques behind Self-Driving Cars, across the full stack of a vehicle's autonomous capabilities. To begin, you'll learn how to apply Computer Vision and Deep Learning toward Perception problems like lane finding, classifying traffic signs, as well as a full end-to-end algorithm for driving with behavioral cloning. You also learn how to track objects from radar and lidar data with sensor fusion. From there, you'll learn and implement the concepts behind Localization, Path Planning and Control, making sure your vehicle knows where it is in the environment and how to navigate through it. At the end of the program, all of these skills will be combined into an integrated capstone, with your code run on a real self-driving car.

Prerequisite Skills

A well-prepared learner is able to:

- Students should have experience with Python, C++, Linear Algebra, and Calculus.
- Intermediate Python (Classes, Data structures)
- Intermediate C++ (Classes, Memory management, Linking)
- Basic Linear Algebra (Matrices, Vectors, Matrix multiplication)
- Basic Calculus (Derivatives, Integrals)
- Basic Statistics (Mean, Standard deviation, Gaussian distribution)
- Basic Physics (Forces)

Required Software

- Python 3.4 or higher
- Anaconda 4.7 or latest
- Jupyter 6.0.1 or latest
- GitHub
- TensorFlow
- AWS Regular acct. with CC - GPU requirement
- IDE
- OpenCV
- EFK Simulator

- uWebSocketIO
- Docker Container
- Ubuntu
- VirtualBox

Version: 3.0.0

Length of Program: 164 Days*

** This is a self-paced program and the length is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.*

Part 1: Computer Vision, Deep Learning, and Sensor Fusion

Here, you'll first become an expert in applying Computer Vision and Deep Learning on automotive problems. You will teach the car to detect lane lines, predict steering angle, and more all based on just camera data, along with working with lidar and radar data later on.

Project: Finding Lane Lines

Write code to identify lane lines on the road, first in an image, and later in a video stream.

Supporting Lessons

Lesson	Summary
Computer Vision Fundamentals	Get a taste of some basic computer vision techniques to find lane markings on the road.

Project: Advanced Lane Finding

Write a software pipeline to identify the lane boundaries in a video from a front-facing camera on a car.

Supporting Lessons

Lesson	Summary
Camera Calibration	Learn how to calibrate your camera to remove inherent distortions that can affect its perception of the world.
Gradients and Color Spaces	Learn how to use gradient thresholds and different color spaces to more easily identify lane markings on the road.
Advanced Computer Vision	Discover more advanced computer vision techniques to improve upon your lane lines algorithm!

Project: Traffic Sign Classifier

Put your skills to the test by using deep learning to classify different traffic signs!

Supporting Lessons

Lesson	Summary
Neural Networks	Build and train neural networks from linear and logistic regression to backpropagation and multilayer perceptron networks.
TensorFlow	The Principal Scientist at Google Brain introduces you to deep learning and Tensorflow, Google's deep learning framework.
Deep Neural Networks	Learn how to go from a simple neural network to a deep neural network.
Convolutional Neural Networks	Learn the theory behind Convolutional Neural Networks and how they help us dramatically improve performance in image classification.
LeNet for Traffic Signs	Using the infamous LeNet neural network architecture, take your first steps toward building a Traffic Sign classifier!

Project: Behavioral Cloning

Train a deep neural network to drive a car like you!

Supporting Lessons

Lesson	Summary
Keras	Take on the neural network framework, Keras! Build and train neural networks more easily.
Transfer Learning	Learn about some of the most famous neural network architectures, and how you can use them to create new models by leveraging existing canonical networks.

Project: Extended Kalman Filters

Apply everything you've learned about Sensor Fusion by implementing an Extended Kalman Filter in C++!

Supporting Lessons

Lesson	Summary
Sensors	Meet the team at Mercedes who will help you track objects in real-time with Sensor Fusion.
Kalman Filters	Sebastian Thrun will walk you through the usage and concepts of a Kalman Filter using Python.
C++ Checkpoint	Are you ready to build Kalman Filters with C++? Take these quizzes to find out!
Geometry and Trigonometry Refresher	This optional content is designed to refresh knowledge of trigonometry and geometry in support of Term 1 objectives.
Extended Kalman Filters	Build a Kalman Filter in C++ that's capable of handling data from multiple sources.

Part 2: Localization, Path Planning, Control, and System Integration

Here, you'll expand on your sensor knowledge to localize and control the vehicle. You'll evaluate sensor data from camera, radar, lidar, and GPS, and use these in closed-loop controllers that actuate the vehicle, finishing by combining all your skills on a real self-driving car!

Project: Kidnapped Vehicle

In this project, you'll build a particle filter and combine it with a real map to localize a vehicle!

Supporting Lessons

Lesson	Summary
Introduction to Localization	Meet the team that will guide you through the localization lessons!
Markov Localization	Learn the math behind localization as well as how to implement Markov localization in C++.
Motion Models	Learn about vehicle movement and motion models to predict where your car will be at a future time.
Particle Filters	Sebastian will teach you what a particle filter is, as well as the theory and math behind the filter.
Implementation of a Particle Filter	Now that you know the theory, learn how to code a particle filter!

Project: Highway Driving

Drive a car down a highway with other cars using your own path planner!

Supporting Lessons

Lesson	Summary
Search	Learn about discrete path planning and algorithms for solving the path planning problem.
Prediction	Use data from sensor fusion to generate predictions about the likely behavior of moving objects.
Behavior Planning	Learn how to think about high-level behavior planning in a self-driving car.
Trajectory Generation	Use C++ and the Eigen linear algebra library to build candidate trajectories for the vehicle to follow.

Project: PID Controller

Implement a PID controller in C++ to maneuver the vehicle around the lake race track!

Supporting Lessons

Lesson	Summary
PID Control	Learn about and how to use PID controllers with Sebastian!

Project: Improve Your LinkedIn Profile

Find your next job or connect with industry peers on LinkedIn. Ensure your profile attracts relevant leads that will grow your professional network.

Project: Optimize Your GitHub Profile

Other professionals are collaborating on GitHub and growing their network. Submit your profile to ensure your profile is on par with leaders in your field.

Project: Programming a Real Self-Driving Car

Run your code on Carla, Udacity's own autonomous vehicle!

Supporting Lessons

Lesson	Summary
Autonomous Vehicle Architecture	Learn about the system architecture for Carla, Udacity's autonomous vehicle!
Introduction to ROS	Obtain an architectural overview of the ROS Framework and setup your own ROS environment on your computer.
Packages & Catkin Workspaces	Learn about ROS workspace structure, essential command line utilities, and how to manage software packages within a project.
Writing ROS Nodes	Learn to write ROS Nodes in Python.

Udacity

Generated Mon May 11 17:24:20 PDT 2020