SeDriCa Freshie Training Modules

Computer Vision:

SeDriCa CV Freshie Training : Although the initials days may seem kind of boring since you are repeatedly watching lectures however I would like to warn you that to progress in a field as machine learning you need a strong background of many ideas floating around without which you would be wandering on a barren island when trying to solve any engineering problem. Each idea or concept you read consider it as a seed in the forest of ideas you are creating.

Objectives :

1. Decent proficiency in designing and implementing an ML model tuned to the purpose
2. Knowledge of basic image processing techniques

Resources : CV training Resources

Basic Structure/ Training Plan :

Day 1:

- □ Week 1 of Reference Course 1
- □ Week 2 of Reference Course 1

Day 2:

- □ Week 3 of Reference Course 1

- □ Week 4 of Reference Course 1

Day 3:

- □ Week 1 of Reference Course 2
- □ Week 2 of Reference Course 2

Day 4:

- □ Week 3 of Reference Course 2
- □ Watch Lecture 5 of Stanford Course CS231N
- □ Catch up with lost work

Day 5:

- □ Watch lectures 9-11 of Stanford CS231N
- □ [OPTIONAL] Lecture 13 (GAN) and Lecture 14 (RL) of stanford CS231n

Day 6 :

- □ Replicate the results stated in “Depth Estimation from Single Image Using CNN-Residual Network”

Day 7:

- □ Read blog post about region proposal networks - blog 1
- □ Read blog post about single shot detectors - blog 2

Day 8:

- □ Replicate the results stated in “Self-Driving Car Steering Angle Prediction Based on Image Recognition”

Decision Making :

Welcome to the Decision Making Subsystem! As a newbie, it’s important to get familiar with what decision making actually does here at SeDriCa. So first thing is first, let’s just take a look at how the foundation of a Self Driving Car is laid.

- **Overall Framework** - To understand the overall layout of how the components and working stages of an Autonomous Car are selected and deployed, go through the following Research paper - https://drive.google.com/file/d/1O3HBN7hYS5d3TC8Y-TejgdRa4-GHFyZx/view?usp=share_link

  As a first task, you must go through the research paper and make notes about the same that shall be reviewed in future meets (**NOTE** : Do not take this as a college academic activity and make unnecessarily long notes, the point I am trying to get across is how the pieces of the Self Driving Car Puzzle fit together and the Skill of reading Research Papers)

Now that you have an understanding of what all should be there in a self Driving Car, let's see how Decision Making fits in this grand scheme of things. Have you heard of **Finite State Machines**?

- **Finite State Machines**: https://drive.google.com/file/d/1Ub8PbIQ6vGfliWGxd2-LSeuONKp7WTQ6/view?usp=sharing

As the second task, you must go through this paper and understand what a Finite State Machine is and how it is implemented. Don't get too worked up if it seems a little confusing right now, soon things will start falling in place, and if it seemed too simple, remember that this simple idea is the fundamental basis of Reinforcement Learning, that is something I promise.

How does a Finite State Machine work with a car though? For that, you need to speak to Professor Steven Waslander of the University of Toronto (don't actually try to contact him please, Joke tha)

- **Behavioural Planning for Autonomous Vehicles**: To understand the implementation of a Finite State Machine as a first Decision Making Step in Autonomy, here's your first stepping stone : https://www.coursera.org/learn/motion-planning-self-driving-cars

In this part, your job is to go directly to just take up the free version of the course, don't need to pay to learn shit ever, and skip over directly to Week 5 and go through the lectures as well as the Supplementary Reading Module(honestly, this shall help you improve your Research Paper reading skills as well as Behavioural Planning Skills). Make notes of what you learn and we can discuss them over a meet.

To familiarize yourself with the way all these things finally come together at SeDriCa, I need you to go through the following reports and Make notes of how you think SeDriCa as a team actually puts all of this together and if you come up with any ideas, they're all more than welcome, and in fact Waffle treat incoming if you give me something to blow my mind, That is something I promise.

- **Report 1** - https://docs.google.com/document/d/1S8JPmeq3dYXXLho_TiOTP2eHu2RpRE3dDMowyehDGGg/edit?usp=sharing
- **Report 2** - https://docs.google.com/document/d/141PIm2U3V1xJXdhoPwY_-k0j98Zpv3x_1AN-yd9oD1o/edit?usp=sharing

(Part 2 of Training module shall soon be put up)

Localisation :

SLAM (1 week):

The two major tasks of this subsystem is to localise your vehicle and map the surroundings of the vehicle. The SLAM problem is to perform these two tasks simultaneously.

The history of SLAM:

https://youtu.be/x5CZmlaMNCs

(Just skim through the video to understand how this problem originated)

Another introduction:

https://www.youtube.com/watch?v=0I30M6yTklo

There are two common approaches to SLAM, one using LiDAR and another using Cameras (also known as Visual SLAM or VSLAM).

A paper on LiDAR based SLAM:

https://www.remedypublications.com/open-access/a-survey-of-slam-research-based-on-lidar-sensors-4870.pdf

EKF (1 week):

Localisation subsystem uses various sensors to determine the state of the vehicle. However, the data received through these sensors contain noise. To minimise the noise and get an accurate result, we use filtering algorithms like EKF.

An introduction to EKF:

https://youtu.be/DE6Jn2cB4J4

https://www.youtube.com/watch?v=XeWG5D71gC0

A ROS package known as robot_localization is one of the most popular packages available for EKF and we are currently using the same. The documentation of the package:

http://docs.ros.org/en/noetic/api/robot_localization/html/index.html

Try to understand how this package works. We use this package to fuse data of GPS, IMU and speedometer (its data is converted to odometry first) sensors.

Popular SLAM algorithms (1 week):

In most of the SLAM algorithms there are some common techniques used. To understand the same, go through two of the most popular SLAM solutions:

LeGO LOAM documentation (A popular LiDAR SLAM package):

https://github.com/RobustFieldAutonomyLab/LeGO-LOAM/blob/master/Shan_Englot_IROS_2018_Preprint.pdf

ORB SLAM 2 documentation (A popular VSLAM package):

https://arxiv.org/pdf/1610.06475.pdf

If this gets a little difficult to understand, here is a review paper, make sure that you read this one thoroughly:

https://medium.com/@sallyrobotics.blog/a-review-on-orb-slam-2-paper-3554d4fcaa7c

And that's it for your training! Call yourself a Jr. Localisation Engineer now.

Motion Planning :

1. Data Structures and their implementation in C++:
    a. STL C++
    b. Introduction to Graphs
    c. Graph Representation in C++
    d. Graph Traversals :
        i. BFS
        ii. DFS
2. Path Finding Algorithms and Related Research Papers
    a. Autonomous Vehicles on The Edge
    b. A tutorial on BFS & DFS
    c. A*algorithm
    d. Model Predictive Control (MPC) - needed for merging controls and Motion Planning

Controls :

MPC training:

1. https://www.youtube.com/watch?v=oBc_BHxw78s&list=PLUMWjy5jgHK1NC52DXXrriwihVrYZKqjk
   Watch the following playlist:
   1-9, 15-21, 26-42

   [1.25 weeks]

2. Understanding PID Control, Part 1: What Is PID Control? - YouTube To get basic controls ka understanding watch this
   Also, watch parts 2 and 3
   [3 days]

3. Understanding Model Predictive Control - MATLAB & Simulink (mathworks.com)
   Basic MPC understanding (view the 4 videos) [4 days]

4. FAQ — do-mpc 4.0.0 documentation
   The library we are currently using. Explore Karlo doc ko Thoda(simple hi hai), and try to code up one of the examples given. (Can skip the graphics vala part) [1-2 days]

5. Lecture Notes | Dynamic Systems and Control | Electrical Engineering and Computer Science | MIT OpenCourseWare

6. Controls - Google Drive
 Ismein 4 vid hai, meets with the prof these can be seen for a good understanding

7. Nonlinear Model Predictive Control_ Theory and Algorithms [Grüne & Pann... Last mein can look at the first 3 chapters of this book(fairly technical toh can be done slowly.