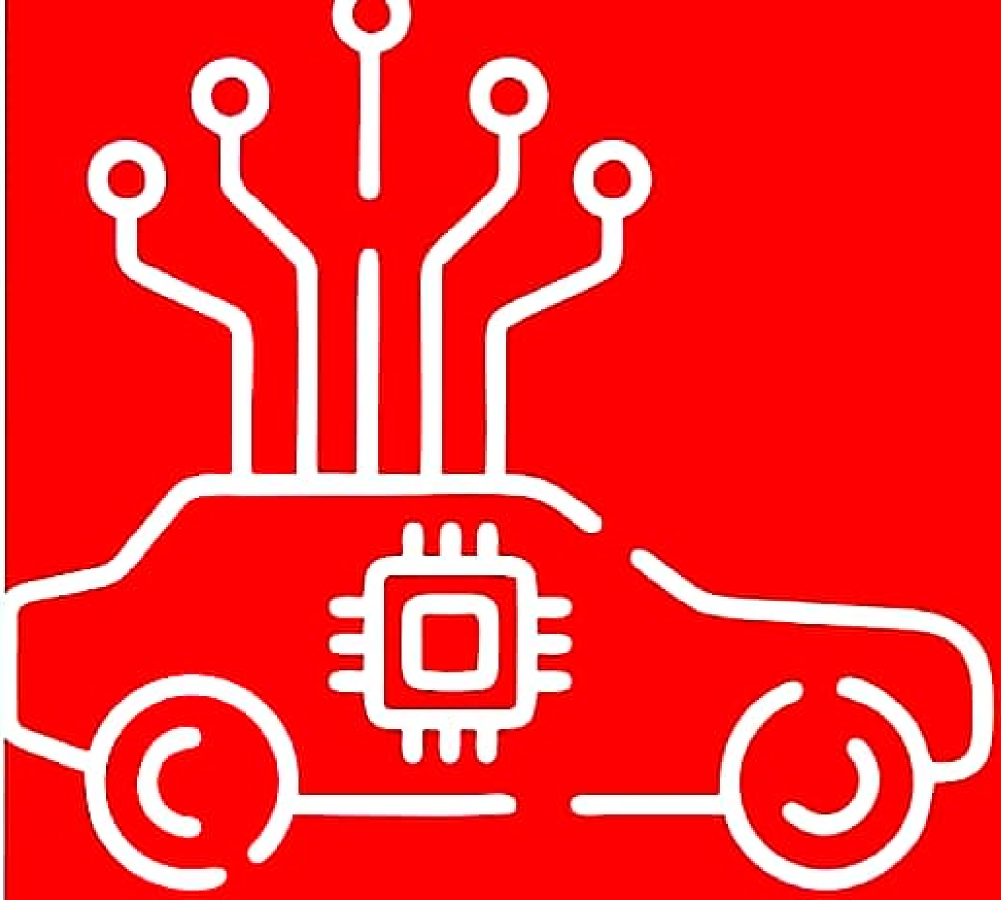
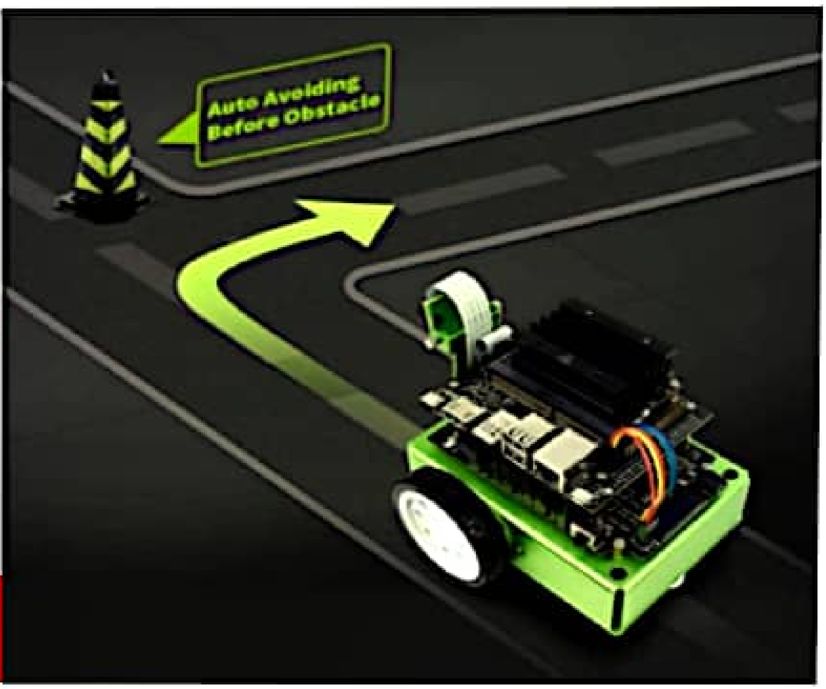


DEEP LEARNING

SELF-DRIVING ROBOTIC CAR ON A SHOESTRING BUDGET

AN OVERVIEW OF HOW TO BUILD A RASPBERRY PI AND TENSOR FLOW POWERED SELF-DRIVING ROBOTIC CAR



INTRODUCTION

Today, Tesla, Google, Uber and GM are all trying to create their own self-driving cars that can run on real-world roads. Many analyst predict that within next 5 years, we will start to have fully autonomous cars running in our cities, and within 30 years, nearly all cars will be fully autonomous. Wouldn't it be cool to build your very own self-driving car using some of the same techniques the big guys use? Let's build our own physical, deep learning, self-driving car from scratch. You will be able to make your car detect and follow lanes, recognize and respond to traffic signals and people on the road in under a week.

ROAD MAP

PART - 1

Introduction to hardware and how to set them up

Mounting of Raspberry Pi and other components

Installation of software drivers needed by raspberry pi and pi cam

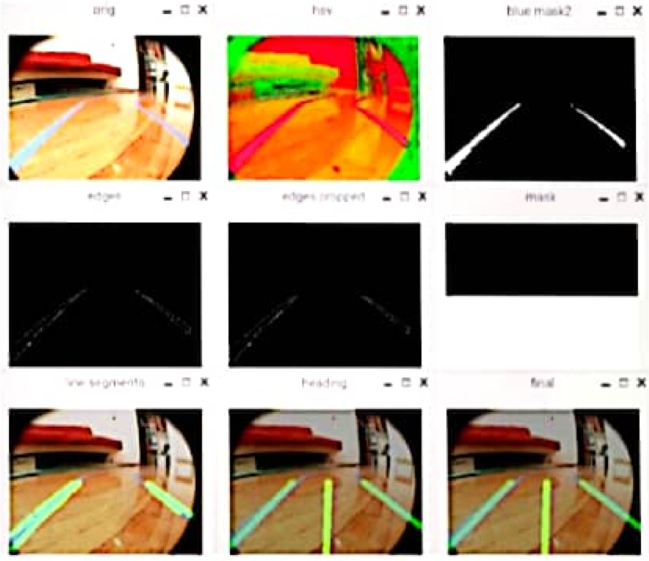
Set up of computer vision and deep learning software needed

The main software tools we are Python (the de-facto programming language for Machine learning AI task), Open CV (a power computer vision package) and Tensor flow (Google's popular deep learning framework). All the software we use here are free and open source

With the (tedious) hardware and software setup out of the way, we will drive right into fun part!

PART - 2

Use python and Open CV to teach DeepPiCar to navigate autonomously on a winding single lane road detecting lane lines and steer accordingly

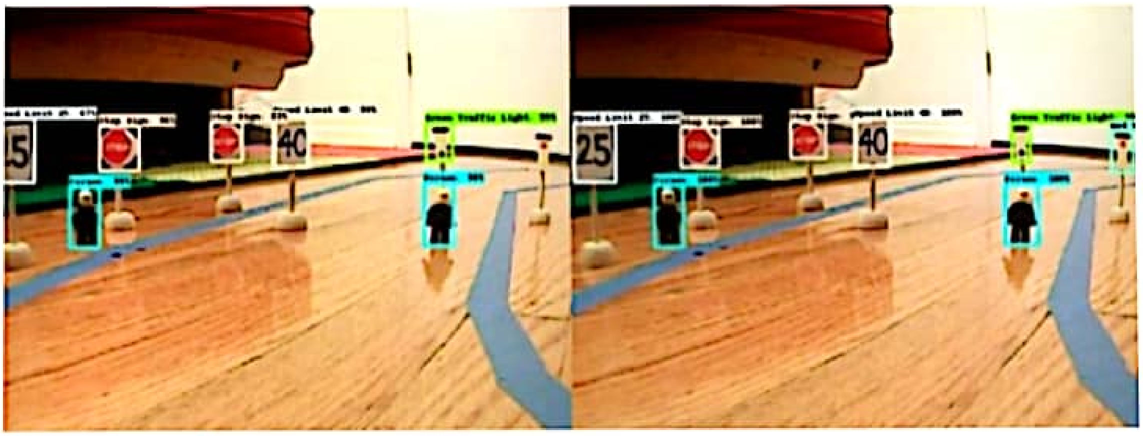


PART - 3

We will train DeepPiCar to navigate lane autonomously without having explicitly write logic control it; as we done in our part 1. This is achieved by "behaviour cloning", where we use just the videos of the road and correct the steering angles for each video frame to train DeepPiCar to drive itself.

PART - 4

We will use deep learning techniques such as single shot multi-box detection and transfer learning to teach DeepPiCar to detect various (miniature) signs.



PREREQUISITE

Basic python programming skills

Basic Linux operating system knowledge, I will assume you know how to run commands in Bash Shell Linux, which is raspberry pi's operating system