# Capstone 2 - Project Ideas

**Idea1: Handwriting Recognition (Classifying Handwritten Text)**

Source: https://www.kaggle.com/landlord/handwriting-recognition

Dataset: The kaggle dataset for the handwritten recognition is a huge dataset with more than four hundred thousand handwritten images of names. There are 206,799 first names and 207,024 surnames in total. The data was divided into a training set (331,059), testing set (41,382), and validation set (41,382) respectively and 3 csv files that contains the transcribed handwritten name.

Thoughts: The idea here is to explore the task of classifying handwritten text and to convert handwritten text into the digital format. This could be in future used to develop an app that could convert handwritten scripts to documents.

**Idea 2: Tesla Stock Price Prediction**

Source: <https://www.kaggle.com/salehahmedrony/tesla-latest-stock-data-2010-2020>

Dataset: The kaggle dataset consists of the latest stock data of Tesla until 2020 that is compiled from Yahoo finance. The dataset has 2417 rows and 7 columns – Date, Open, High, Low, Close, Adj Close, Volume.

Thoughts: To predict the future Tesla stock price.

**Idea 3: Traffic sign Recognition (Deep Learning - CNN)**

Source: https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

Dataset: The German Traffic Sign Benchmark dataset has More than 40 classes, More than 50,000 images in total

**Idea 4: Self Driving Car**

**Source:** [**https://github.com/udacity/self-driving-car**](https://github.com/udacity/self-driving-car)

[**https://github.com/microsoft/AirSim**](https://github.com/microsoft/AirSim)

As a part of Self Driving Car project,

1.Lane Detection

<https://github.com/rslim087a/Self-Driving-Car-Course-Codes/tree/master/Section%205%20Resources%20(Finding%20Lanes)/Image%20Needed%20for%20Loading%20Image%20Lecture>

2.Classifying Road Symbols using German Traffic Sign Dataset

3.Speed control and throttle control using the open source simulator available on Udacity platform.

Simulator interface (C# / Unity)

The simulator interface starts up with the car to be controlled positioned on a two-way six-lane highway with other cars passing by. There is a button that enables the “Manual Mode”, in which the user can control the car manually using the arrow keys. The project uses the RoadArchitect library [19], that is able to generate complex roads, bridges and tunnels in Unity, and the SocketIO library for Unity3d [SocketIO] library for WebSockets communications.