Vehicle Detection using Machine Learning

Domain Background

Humans. We are highly social and emotional creatures. With repetitive tasks, we often get bored, and our efficiency begins to reduce when we work for long hours. We don't like to be confined to something for very long.

With the growing size of the world population and the rising demand for better and faster services, there is a need for faster and more efficient ways of doing things. What better way than to use machines for the job!? And that's precisely why automation exists.

Modern machines are highly complex tools that have made automation possible. Automation has changed the way people work and live. It helps us with tasks that are often repetitive, boring or even dangerous in nature—so that we may spend our time in more meaningful pursuits. With the advent of industrial revolution, the steam engine and ultimately the computers—laborious tasks which used to require many hours of human effort are now being done much more efficiently and quickly by machines.

The research and implementation of **autonomous vehicles**, in particular, has an immense impact in our everyday lives. According to wikipedia—An autonomous vehicle, or a self driving car is a vehicle that is capable of sensing its environment and navigating without human input. Just like any other machine, a self-driving car has many components that work together to transport people and goods safely from one place to another.

Autonomous vehicles have the potential to transport people and goods much more safely than human drivers, and also on a much larger scale. There are plenty of challenges and problem areas that need to be addressed in order to achieve this objective. With the latest advances in computing power, it's become possible for us to implement various mathematical and scientific research in solving these problems.

Problem Statement

It's important to ensure that an autonomous vehicle detects the presence of other vehicles on the road—so that it does not pose a danger to itself or to other vehicles around it. In this Capstone project, we explore an area that is paramount to ensuring safety in autonomous vehicles—**computer vision**.

Datasets and Inputs

We use two different sets of data for our purposes. The first data-set consists of vehicle images, and the other data-set consists of non-vehicular objects on the road. We need both these data-sets so that our agent can learn to distinguish between vehicles and non-vehicles on the road.

For recognizing vehicles, the vehicle data-set has 4 different categories of images which show vehicles ahead of our agent from different viewing angles and distances. These images will be used to train our agent to recognize vehicles towards its left side and right side, vehicles that are far, and the ones that are closer.

The second data-set consists of images of objects other than vehicles that can be seen on the road. These could include trees, pavement markings, side-walks, traffic signs, road dividers, the sky in various times of the day, and so forth.

Solution Statement

We use computer vision and machine learning techniques to train our moving car to accurately detect and track vehicles on the road.

Using our training data-sets, it is possible for us to train and test our AI agent, and also evaluate its performance in a live video stream. This is an important component while designing our self driving car.

Evaluation Metrics

We divide the dataset into training and testing subsets. Then, we'll use a technique called Histogram of Oriented Gradients (HOG) to extract features from our training data, and train a SVM classifier (Suppport Vector Machines).

Once we train our model using the data from the training set, we can measure the **accuracy** of our model using images from the validation set. We can experiment with tuning different hyper-parameters to get the best accuracy for our model.

Project Design

First, we'll begin with feature exploration. The process begins with extracting useful information from the image and eliminating noise data, thus making the image simpler. The resulting image is called Feature Descriptor. HOG is a type of feature descriptor that is typically used in our given objective.

Here are the steps to calculate a HOG descriptor—

- We first calculate the horizontal and vertical gradients in the given image.
- Next, we find the magnitude and direction of each gradient using this formula

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

This removes a lot of noisy information from the image, but highlights all the essential outlines.

- Now calculate a Histogram of Gradients by dividing the image by x*y cells. We experiment with the optimum cell size that captures essential information.
- Finally we normalize the gradient vectors, so that the training becomes independent of the brightness of the image, contrast etc.

Using our selected HOG features, we train an SVM classifier. We'll experiment with parameters like the type of kernel used, gamma, color channels etc, and use GridSearchCV to find the optimum hyper-parameter tuning.

Next, we'll use a sliding window search to try and detect vehicles in a live video recording taken from a moving car.