Vehicle and Lane Detection

Rohit Vasudevan

ID: 001352313 | rvasudevan@albany.edu

DEPT. OF COMPUTER SCIENCE AND ENGINEERING,
University at Albany

ABSTRACT: Vehicle and lane detection in digital image and video sequences is one of the key technologies of Intelligent Transportation Systems. Making a quality dataset is crucial for training a good Artificial Intelligence (AI) model for smart and intelligent transportation. In this project, I have taken a dataset comprising of images of vehicles, labelled the vehicles based on their type and trained a linear SVM classifier. I used the SVM classifier to search for vehicles in the images, and for the vehicles detected in the image, a bounding box is estimated. For lane detection in the image, the curvature of the lane was calculated and the lane line spacing was calculated to determine the direction of the lane.

I. INTRODUCTION

Intelligent Transportation System is the application of sensing, analysis, control and communication technologies to ground transportation in order to ensure and improve safety, mobility and efficiency. Many problems such as traffic accidents, traffic congestions, etc have occurred due to the increase in vehicle traffic.

As a result, many investigators have increased their attentions towards Intelligent Transportation Systems, such as to predict the traffic flow to detect congestions, and so forth. To process the information and monitor the result to understand better traffic flow, there is an increasing need for traffic surveillance to understand traffic and why congestion occurs. The detection of vehicles in images and video surveillance data is an extremely challenging problem in computer vision and machine learning with important practical analysis, such as traffic analysis, lane and pedestrian detection and security.

Vehicle labelling and detection is important in basic surveillance systems. The main goal of this project is to identify the vehicles in images and videos and develop a methodology to identify them in the dataset provided.

There is a system that has been developed to help detect vehicles better and more efficiently. It can be done by using either a classifier or by implementing a neural network to help detect vehicles accurately. In this project, I have implemented a linear SVM classifier and performed a HOG feature extraction on a labelled training set of images.

II. LITERATURE SURVEY

The ideas obtained from the below mentioned sections have been used have been cross referenced to indicate the source of the information presented, when needed. This study was very necessary from the point of view of getting the background information to help us proceed in applying the proposed techniques and obtaining the results.

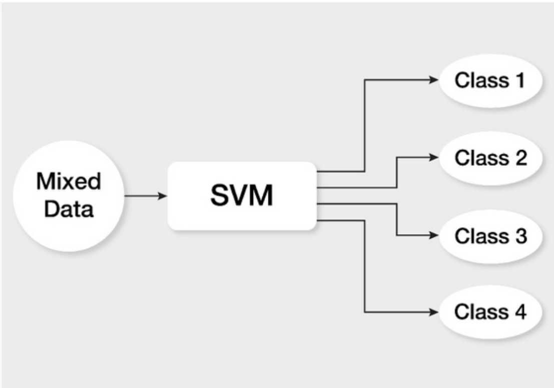
2.1. Machine Learning

Machine learning, a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviours based on empirical data, such as from sensor data or databases. Machine Learning is a scientific discipline that addresses the following question: 'How can we program systems to automatically learn and to improve with experience? 'Learning in this context is not learning by heart but recognizing complex patterns and make intelligent decisions based on data. The difficulty lies in the fact that the set of all possible decisions given all possible inputs is too complex to describe. To tackle this problem the field of Machine Learning develops algorithms that discover knowledge from specific data and experience, based on sound statistical and computational principles.

A learner can take advantage of data to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data, the difficulty lies in the fact that the set of all possible behaviours given all possible inputs is too large to be covered by the set of observed examples (training data). Hence the learner must generalize from the given examples, so as to be able to produce a useful output in new case. We have followed the same principle, where we have designed algorithms to generalize from the given examples and then produce a useful output. Machine learning algorithms are described as either 'supervised' or 'unsupervised'.

2.1.1. Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. They are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis.

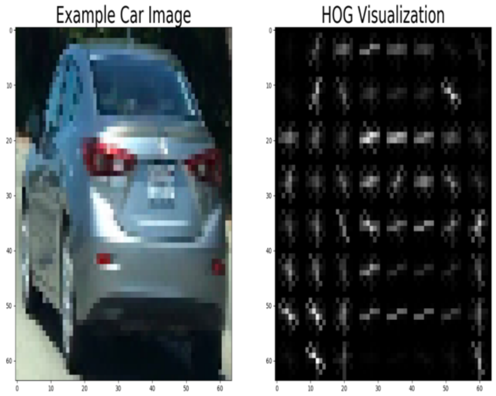


2.2. Histogram of Oriented Gradients

The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the basic purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of images.

The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is the concatenation of these histograms.

The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions.



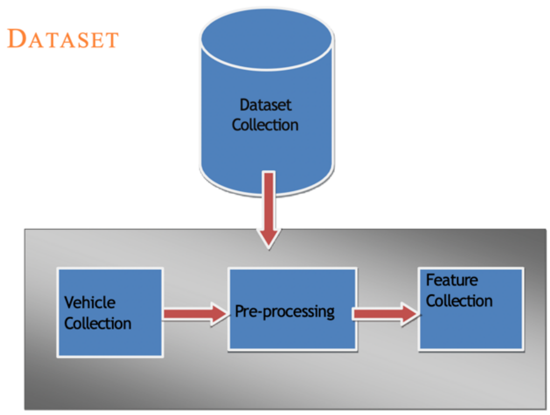
III: Architecture & Methodology

The architecture of the project can be divided into three stages namely :-

1. Dataset Collection
2. Applying the dataset to classifiers
    2.1. Linear SVM (Support Vector Machine)
        2.1.1. Histogram of Oriented Gradients
        2.1.2. HOG Parameters
        2.1.3. Training classifier and sliding window
        2.1.4. Heat Map
3. Neural Network Approach

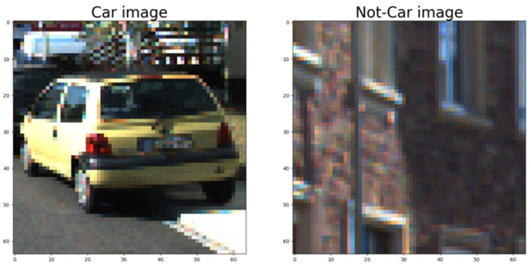
3.1. Dataset Collection

This phase is one of the most important phases of the project development cycle as it is only dataset which defines the level of accuracy and stability that will be achieved and reflected in results. Here we have started with collection of database which comprises of various vehicle images which are labelled as whether they are vehicles or not.



3.1.1. Vehicle Collection

The datasets being considered were the GTI dataset (Grupo de Tratamiento de Imagenes), KITTI dataset. The dataset considered of two lists, which were split into vehicles and non-vehicles.



3.1.2. Extracting features from dataset

For extracting the features from the vehicle dataset, I performed a HOG feature extraction on a labelled training set of images and train a linear SVM classifier. A colour transform is then performed on the image and the binned colour image, HOG feature vector and histograms of colour are appended. The features are then normalized and a selection is done for training and testing.

3.2. Applying Classifiers and Analysis

We have a dataset of over 8000 images as vehicles and 8000 images as non-vehicles.

3.2.1. Linear SVM

Scikit package in Python offers a variety of options for implementing SVMs.
The SVMs require that the features be scaled for both the training and test data, i.e. the mean of the features be 0 and the standard deviation be 1. So, we had to pre-process our data before feeding it to the SVM.

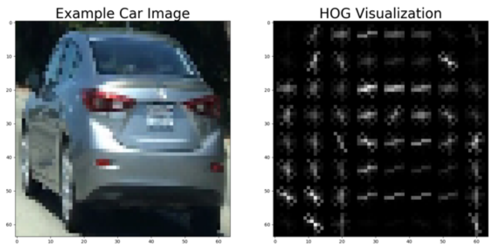
In this project, I have performed a histogram of oriented features (HOG) feature extraction on a labelled set of training images. The features are then normalized, and a selection is randomized for training and testing. I then implemented a sliding window technique and used the trained classifier on the images to search for vehicles. For the vehicles that are detected in the images, a bounding box is estimated.

3.2.1.1. Histogram of Oriented Gradients

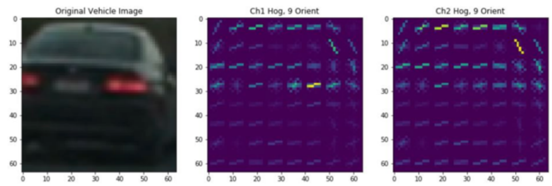
The first step was to extract the histogram of oriented gradients from the training images. After reading the images from the dataset, different colour spaces and different skimage.hog() parameters such as orientation, pixels per cell and cells per block were explored. To optimize the HOG extraction, the HOG feature is extracted only once for the entire image and the entire image is saved for further processing.

The below images show the RGB and HLS color space and the HOG parameters considered. For the RGB model, orientations = 9, pixels per cell = (8,8) and cells per block = (2,2). For the HLS model, orientations = 9 & 12, pixels per cell = (8,8) and cells per block = (2,2).

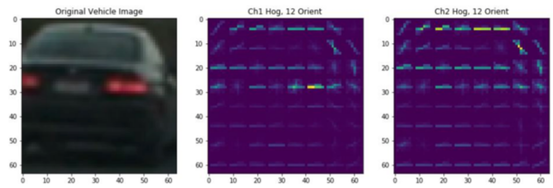
RGB:



HLS:
Orientation = 9



Orientation = 12



3.2.1.2. HOG Parameters, Spatial Features

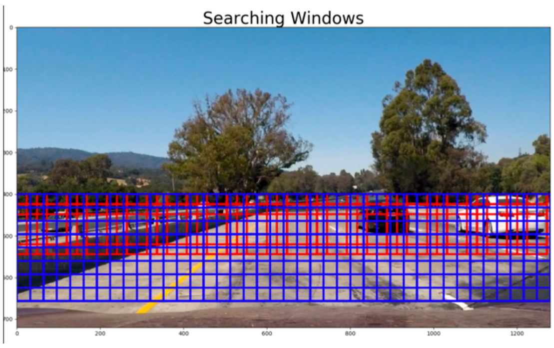
Determining the final choice of HOG parameters was based on the following terms:
1. SVM Model Test Accuracy
2. Positive Detections
3. Detection Stability
4. False Positive and False Negative Detections

The various parameters that are considered are:
1. YCrCb color space.
2. orientation = 9
3. pixel per cell = 8
4. cell per block = 2
5. hog channel = ALL
6. spatial size = (32,32)
7. hist_bins = 32
8. spatial feat = True
9. hist feat = True
10. hog feat = True

3.2.1.3. Training classifier and sliding window

I randomly select 20% of images for testing and others for training, and a linear SVM is used as classifier.

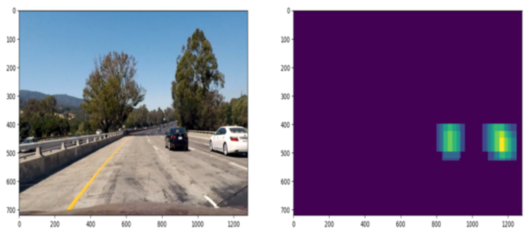
For this SVM-based approach, I use two scales of the search window (64x64 and 128x128, see line 41) and search only between [400, 656] in y axis. I choose 75% overlap for the search windows in each scale. For every window, the SVM classifier is used to predict whether it contains a car nor not. If yes, save this window. In the end, a list of windows contains detected cars are obtained.



3.2.1.4. Heat Map

After obtained a list of windows which may contain cars, a function named generate_heatmap is used to generate a heatmap. Then a threshold is used to filter out the false positives.

For image, we could directly use the result from the filtered heatmap to create a bounding box of the detected vehicle.

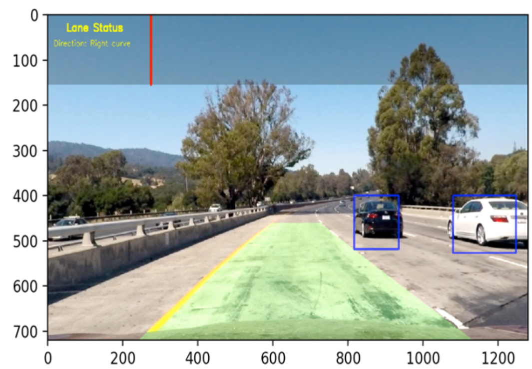


3.3. Neural Network Approach

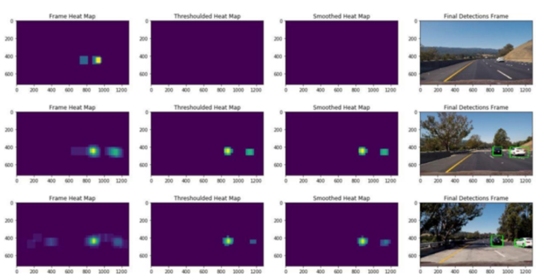
YOLO is an object detection pipeline based on Neural Network. Contrast to prior work on object detection with classifiers to perform detection, YOLO frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

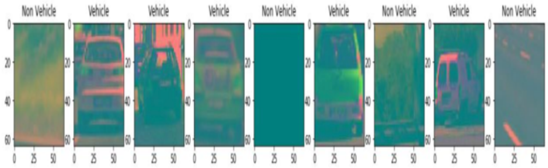
IV. Results

4.1. SVM Pipeline



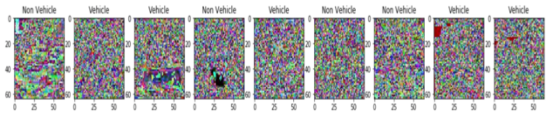
4.1. Final Detection Pipeline using YCrCb color space

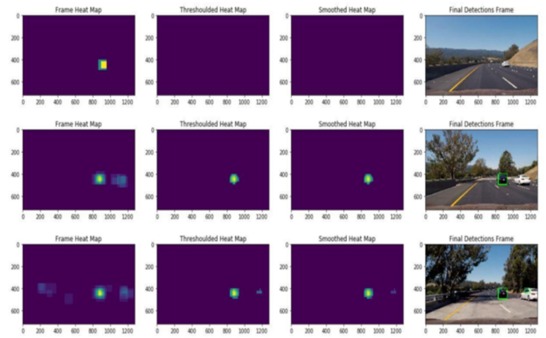




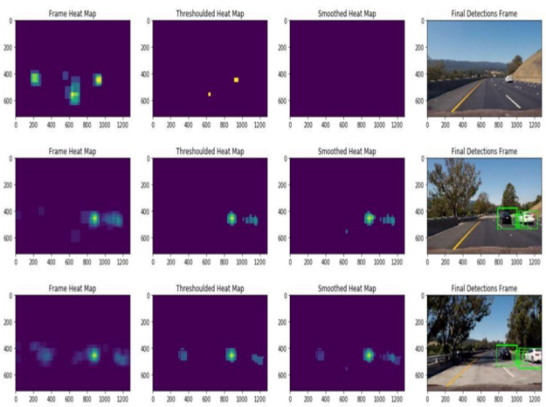
4.2. Results of other color models

4.2.1 LUV color space

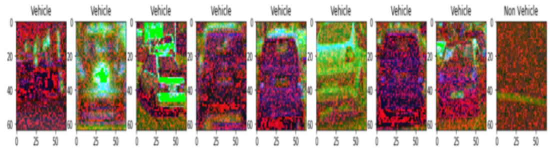


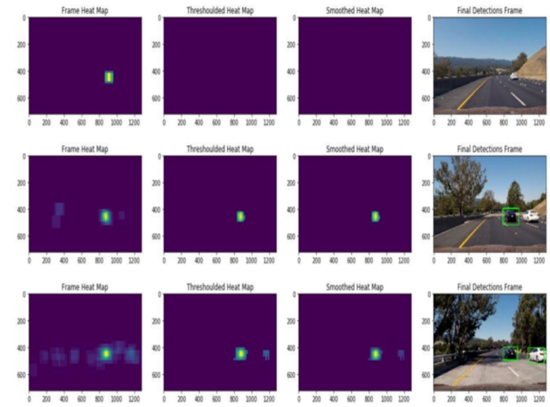


4.2.2 RGB color space



4.2.3 HLS color space





V. Conclusion and Future Work

The Linear SVM approach can accurately detect the cars present in the image and estimate a bounding box for the image and the lane present. The lane direction is also estimated accurately.

Some of the observations I noticed was that, the model was performing better for the black car detection compared to the other car colour detection, and by adding more data or probably by augmenting the light images in the same data, a better performance for the model can be obtained.

VIII. References

[1] http://www.gti.ssr.upm.es/data/Vehicle_database.html

http://www.cvlibs.net/datasets/kitti/

[2] Udacity – CarND Vehicle Detection.