Vehicle Detection Project

Author: Ashish Vishwakarma (Student)

- Extracting Histogram of Oriented Gradients (HOG) features and color features on a labeled training set of images and train a linear SVM classifier
- Implement a sliding-window technique and use trained classifier to search for vehicles in images.
- Run your pipeline on a video stream (start with the test_video.mp4 and later implement on full project_video.mp4) and create a heat map of recurring detections frame by frame to reject outliers and follow detected vehicles.
- Estimate a bounding box for vehicles detected.

[Rubric](#)

Rubric Points

Here I'll consider the rubric points individually and describe how I addressed each point in my implementation.

Writeup / README

In this file, I'll describe my approach to the project, and show the results of each stage. The code for this project is contained in the IPYB file "Vehicle_Detection.ipynb" and the output video files are "project_video_output.mp4" and "test_video_output.mp4".

Table of Contents

[Rubric Points](#)
[Writeup / README](#)
[Table of Contents](#)
[Import Packages](#)
[Import Data Set](#)
[Exploratory Data Analysis](#)
[Visualization of Random Images](#)
[Histogram of Oriented Gradients (HOG)](#)
[Training Classifier](#)
[Sliding Window Search](#)
[Video Implementation](#)
[Discussion](#)

IMPORTING PACKAGES

```
%matplotlib inline
import os
import glob
import time
import numpy as np
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import cv2
from skimage import color, exposure
from sklearn.svm import LinearSVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from moviepy.editor import VideoFileClip
from IPYTHON.DISPLAY import HTML
```

LOADING THE DATA

```
# Define a function to return some characteristics of the dataset
def data_look(car_list, notcar_list):
    data_dict = {}
    # Define a key in data_dict "n_cars" and store the number of car images
    data_dict["n_cars"] = len(car_list)
    # Define a key in data_dict "n_notcars" and store the number of notcar images
    data_dict["n_notcars"] = len(notcar_list)
    # Read in a test image, either car or notcar
    example_img = mpimg.imread(car_list[0])
    # Define a key in data_dict "shape" and store the image shape
    data_dict["shape"] = example_img.shape
    # Define a key in data_dict "data_type" and store the data type of the image
    data_dict["data_type"] = example_img.dtype
    # Return data_dict
    return data_dict
```