Vehicle detection in images using Neural Networks and Color Histogram features - Manual

Jakša Jovičić

June 11, 2020

1 Set-up

The code is written in Python 3.7, in Anaconda environment and the Spyder text editor.

Non standard libraries used are:

- scipy
- opencv
- keras
- tensorflow

Tensorflow and keras are installed following the instruction on this site if you have a NVIDIA gpu: The Best Way to Install TensorFlow with GPU Support on Windows 10 (Without Installing CUDA) If not, keras and tensorflow are installed as any other library using pip install.

2 Data acquisition and cleaning

I used several datasets to train the model well:

1. Comp_cars dataset
2. Motion-based Segmentation and Recognition Dataset
3. Vehicle Image Database
4. Some pictures i took with my phone

The first dataset contains a lot of images of cars together with the bounding boxes for each car, it has images of cars from five viewpoints, Front (F), Rear (R), Side (S), Front-Side (FS), Front-Side (RS). I made a function pictures_from_dataset1() in file data_cleaning.py that goes through all images, crops them according to the labels, resizes them and puts R and F images in one folder and FS and RS images in another folder.

The second dataset are two road camera videos, the function pictures_from_dataset2() in data_cleaning.py goes through frames of both videos and extracts random subimages which are used as not-car class for classification and puts them in a single folder. I went through the result images and deleted some images that I don't want to be in not-car class by hand.

The third dataset contains cropped images of cars from road camera videos from four different positions, as well as the not-car images from same angles.

The fourth dataset is made from many pictures i took with my phone in different occasions, function pictures_from_dataset4() in data_cleaning.py extracts random subimages from every image and puts them in one folder to be used for not-car class.

So all that should be done is:

- Download all datasets
- Put them in separate folders named dataset1-4 and all those folders in one folder which will be root
- Run data_cleaning.py with root = complete path to the root folder

3 Feature extraction

Once the data_cleaning.py has been run, the pictures have been extracted from all datasets and are ready for feature extraction. Function feature_extraction_all() in feature_extraction.py goes through all folders with pictures and extracts the features and puts them in separate files for each class and each dataset.

The features that are being used are bin-split color histograms of each image, 3 histograms with 256 elements are spit in 64 bins each and they are concatenated in one list that is used as the NN input.
I found that the model works better when I don't use FS and RS features for training so they are not used.

Function group_features() in feature_extraction.py groups all features from different folders in two files which will be used for training.

So all that should be done is:

- Chose which features you want to use for training (default is all except FS and RS)
- Run feature_extraction.py

4 Model training

Model is trained by running the training() function in feature_extraction.py, it will create text files representing the parameters of the trained network.

5 Testing

Python script test.py is used for model testing, it has functions image_search_full(), folder_search() and video_search() that do the whole processing on an image, a folder of images and a video.

6 Further work

- Add more data, datasets with more labeled car pictures from surveillance should make the model work a lot better, because my model is trained on images that are almost all from car shows and taken with professional cameras and not with road cameras.

- Combining this model with some other model could give better results
- Make several models and a voting system for detection
- Better search system e.g. prediction based on the previous frame for videos
- Optimization, speed up
- The system is highly dependant on decision threshold, so some kind of adaptive threshold could make the system work better