Traffic Density Estimation using YOLO

We have used dark flow implementation of YOLO that can be downloaded from the below link.

https://github.com/thtrieu/darkflow

Make sure that dark flow and TensorFlow is properly installed in your system for the project to run.

This project has 4 python files and each file performs different roles:

1.flaskk.py:

This is the first file that is called by the user. It is used to get the data from the user like path of the input video, coordinates of the line and the time of the video. This file first checks the input parameters and then calls another file called mm.py.

2.mm.py

This file has two function which are called by the above file. The first function called the convert () function firsts takes the path of the input video as an argument and then splits the video into different videos having constant time interval of 10 minutes.

This method after detection makes a new row in the file called count.csv and writes the number of cars, bike, person and other vehicles detected along with the time.

The other function is called graph () function. This function is used to make a graph of the count of different vehicles detected by the algorithm.

3.main2.py

This file is responsible for the detection of objects in the video. We have saved a pre-trained custom model of YOLO in the build_graph file and we are using that model instead of training a new model again and again in order to make the algorithm faster. Make sure that the path to these files are correct. This file is the crux of our project.

This file has two functions: first is boxing () function. This function takes the list of predictions and if the confidence of the prediction is greater than 40% then it draws a bounding box around that object and classifies that object.

Other function is vehicle () function. This function processes the video, calls the boxing () function on each frame and also keeps the count of different vehicles and displays it.

4.Sort.py

This is the file which performs the counting operations on the detected objects. The main2.py file passes the bounding box coordinates to this file and this python file performs the counting and gives us back the count of all the vehicles which is displayed by the previous file on the video. This file was originally made by Alex Bewley.