DC-DRIVE COMPANION

Abstract

Driving is a very dangerous activity, both for the people inside and outside vehicles. This is because humans are fallible. For this reason, the analysis of the driving risk is an important matter, both for the driver and for public safety in general.

The goal of this project was to develop a framework to analyse the risk of a person's driving style using Machine Learning techniques so that the application would improve its accuracy and use cases range over time. The analysis was done using the inertial sensors of a smartphone and the On-Board Diagnosis, OBD, to get parameters about the car's real-time state like speed, motor's RPM and engine load. Using this data, it is possible to detect features of the driving which can then be used to classify the user's driving style in one of 3 classes: low, medium and high risk.

Introduction and Motivation

Driving has, for long, been the most dangerous transportation mean of all. In 2016, there was only one general aviation accident involving aircraft with a MTOM (maximum take-off mass) above 2250 kg and in 2013, a specially bad year for aviation accidents (from 2006 to 2013, there had never been more than 5 fatalities per year), air transportation had a total of 11 fatalities while road transportation recorded a mind-boggling number of 25401 fatalities, all in the EU-28 territory.

One of the challenges in this project is the fact that there are no datasets available with all these features and no pre-trained model for this kind of data. This means that all the data that was used had to be gathered and partially labelled by me and one volunteer. Such conditions resulted in a small and unbalanced dataset which crippled the results.

Project Overview

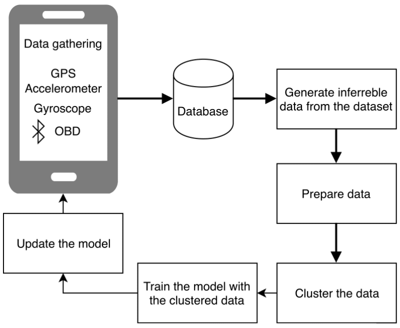
Detecting the Driving Style (DS) risk from a set of raw sensorial and OBD data requires several steps of data pre-treatment, detecting features, labelling, clustering and training the model (not necessarily all or in that order).

FRAUNHOFER AICOS

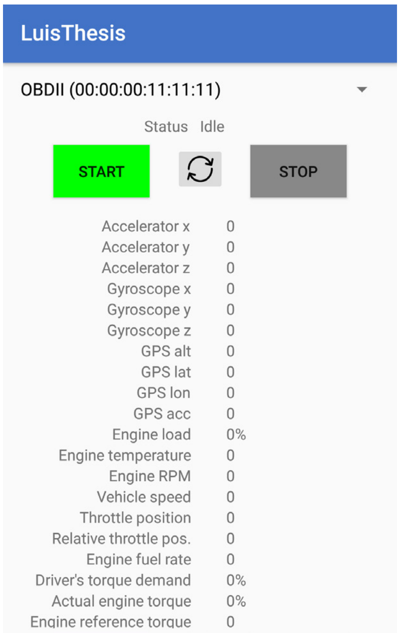
Rua Alfredo Allen, 455 / 460
4200-135 Porto, Portugal

+351 220 430 300
info@fraunhofer.pt
www.fraunhofer.pt

1



2



3

1 *Results of the CNN for turn detection. Yellow points are classified as being in a turn.*

2 *System overview.*

3 *Screenshot of the application in the idle state.*

The diagram that represents the overall process is represented in the Fig. 2. It follows a simplified version of the CRISP-DM methodology.

First, the developed Android App acquires data and sends it raw into the remote DB after each session. Then, when there is enough data in the DB to work with, new data is inferred from the raw data like the standard deviation and average of the measurements in each time frame. Next, clustering algorithms would be used to cluster the data and the one with the best performance will be used to train the model (clustering wasn't used because there was no time and the dataset was small and unbalanced even by the end of the project).

After being clustered, the data would be used to train the model until it reached a maximum in the fitness function. Then, the new model would be uploaded to the App so that it contains the most recent version. The lack of data and the small variation in it, which creates a big challenge in clustering the data into meaningful clusters, was the biggest problem in this project. If the data is scarce, the clusters will not be relevant and possibly even not related to the risk at all.

Data Preparation

The data was aggregated with an SQL query in time-frames with 5 seconds of duration. The time-frames were overlapping with an offset of 1.25 seconds ([0;5[,[1.25;6.25[,[2.5;7.5[,[3.75;8.75[…) to prevent possibility of a key event being sliced into two time-frames. In each time-frame, each data field is averaged and the standard deviation calculated (each column produces two columns). The arithmetic average is important to get the centroid of the data columns in each time-frame, while the standard deviation allows the detection of changes in the values of a data column inside each time-frame. For the purpose of the proof of concept model's dataset, 169 rows of driving data were manually labelled on being on a turn or not. 20 were used in validation and the rest in training.

Results

Because of the small dataset, the final objective of this project wasn't accomplished: the detection of the driving risk. Nevertheless, it was possible to detect a mid-level feature as proof of concept of classifying driving data: turns detection. A fully connected convolutional neural network (CNN) was designed to detect turns and straight road segments in the data. After analysing the graphic, a problem was discovered: turns to the right were correctly classified while turns to the left were often classified as not turns (as represented in Fig. 1). When reviewing the dataset, it was detected that it was mostly composed of right turns, thus justifying this flaw.

Conclusions and Future Work

It wasn't possible to determine the driving risk due to the lack of time and data but it was possible to determine the feasibility of classifying driving events using this method, which hints that it is also possible to determine the driving risk with it. For future work, because the framework was already fully implemented, more data needs to be gathered, labelled and clustered so it's possible to train the model to detect the driving risk.