Report : Drive Safe

Authors

Milan Chaudhari(2015csb1010)
Narotam Singh (2015csb1065)
Shubham Dham (2015csb1117)
Shreya Dubey (2015csb1074)

1 Problem Statement

- To capture the image of the driver at a fixed rate per second.
- To infer from the image whether the driver is distracted or not.
- To alert the driver by some kind of alarm system in case of distraction else keep on capturing images.

2 Description of Solutions and Features

- Drive safe is a stand alone android application that alerts distracted driver by a honking sound.
- Cell phone with the Drive Safe application is fixed on the dashboard near the passenger seat adjacent to the driver.
- It takes images of the driver at the rate of 1 image/second.
- Uses the features associated with the image to infer whether the driver is distracted or not. This inference is based on the data obtained from training on the dataset of the distracted driver obtained from the internet.
- The result so obtained from trained data regarding the clicked image tells whether the driver is distracted or not. All this takes place in the back-end.
- In case of distraction , a honking sound is played on the cell phone.

3 Implementation Details

3.1 Training using Machine Learning

There is a Dataset of 10,000 images on kaggle dataset which is used for training. These images are classified as following:

1. Normal
2. texting - right hand
3. talking on mobile phone -right hand
4. texting âĂŞ left hand
5. talking on mobile phone -left hand
6. operating radio
7. drinking

8. reaching behind
9. hair and makeup
10. talking to passenger

Each image is first converted in 50×50 gray scale image. After that intensity of each pixel is used as a feature to train the dataset above. So the size of input layer is 2500. In Training we have used Neural Networks with 1 hidden layer, with the size of 100. It is an iterative algorithm, where iterations are being done using an optimization algorithm implemented in a function script 'fming.m' file (the file is taken from a machine learning assignment at Coursera Machine Learning). As arguments of fming, we need to pass one initial parameters(which is a matrix of random values), a function that returns the cost and cost gradient and some other options (here it is number of iterations, which is 3000). After running these algorithms, it generates two matrices Theta1 and Theta2 respectively; Theta1 and Theta2 are parameter matrices used to calculate hidden layer from input layer and output layer from hidden layer. After that Theta1 and Theta2 are stored in two '.txt' files, which are used in the file called 'predict.java'.

3.2 Android Application

There are two activities and two java class used in this part :

- MainActivity2
- MainActivity
- Predict.java
- CameraView.java

3.2.1 MainActivity2

The control starts from this activity, which displays a loading bar, which represents how much the data of the two '.txt' files('o1.txt' and 'o2.txt' representing the parameters of neural network hypothesis) is been loaded in two 2-D double arrays.After all of the data is loaded,MainActivity is launched.

3.2.2 MainActivity

All the buttons and an object of CameraView is in this activity. The buttons named 'start', 'pause' and 'resume' are present.The functions of these buttons is clear from their names.MainActivity captures the images and calls the functions of predict class repeatedly.

3.2.3 Predict.java

This is the most important class of the project.After capturing the image ,the image data is passed to the 'prediction' function in predict class where all the operations on the image are done to get the final output classifier.

3.2.4 CameraView.java

Source of CameraView.java: CameraView. This class is being used in MainActivity to embed the camera on the layout below the buttons.

Overall the code flows as follows :

1. Launch application, click on Start button to initialize the process.
2. Images are clicked at the rate of one image per second.
3. Images are scaled to 50×50 , converted to grayscale and converted from JPEG format to RGB format.
4. Features of the final RGB grayscale image are extracted.
5. Features operated with parameters in previously obtained text files from MATLAB, gives the probabilities of occurrence of the different classifications of the dataset.
6. Answer is the class with the highest probability.
7. If same behavior found for three continuous frames , then the driver is considered distracted.
8. Rings alarm for all the cases except the normal.