MACHINE LEARNING ENGINEER NANODEGREE

Capstone Proposal

Chinreddy Subramanyam Reddy
February 28th, 2019

Proposal

Domain Background

- ✓ Distracted driving is any activity that diverts attention from driving, including talking or texting on your phone, eating and drinking—anything that takes your attention away from the task of safe driving.
- ✓ In the last decade alone, India lost 1.3 million people to road crashes and another 5.3 million have been seriously injured. While it has just 1% of the world’s vehicles, India accounts for over 10% of global road crash fatalities.
- ✓ In order to reduce this count of road accidents (due to distracted drivers), we can use machine learning.

Sources:
http://savelifefoundation.org/wp-content/uploads/2017/04/Distracted-Driving-in-India_A-Study-on-Mobile-Phone-Usage-Pattern-and-Behaviour.pdf

Problem statement

The aim of this project is to identify if the driver is distracted or not while driving.

Distraction of driver includes texting, talking to people, using mobile, eating etc. So, this is going to be a classification problem where we classify in which way the driver is distracted while driving.

Datasets and inputs

The data set which I am working on consists of driver images, each taken in a car with a driver doing some activity (texting on mobile, eating something, talking on the phone or with the people who are beside her/him, makeup, reaching behind, etc) were provided.

The target label(class id) has the following 10 different classes:

- ✓ c0: safe driving
- ✓ c1: texting - right
- ✓ c2: talking on the phone - right
- ✓ c3: texting - left
- ✓ c4: talking on the phone - left
- ✓ c5: operating the radio
- ✓ c6: drinking
- ✓ c7: reaching behind
- ✓ c8: hair and makeup
- ✓ c9: talking to passenger

- Each of the above class refers to an activity of the driver which distracts him/her from driving.
- imgs.zip-zipped folder of all (train/test) images.
- driver_imgs_list.csv - a list of training images, their subject (driver) id, and class id
- The dataset has been downloaded from kaggle.
- https://www.kaggle.com/c/state-farm-distracted-driver-detection/data
- There are a total of 10215 images. The images are coloured and have 640 x 480 pixels each as shown in above figures.

Solution Statement

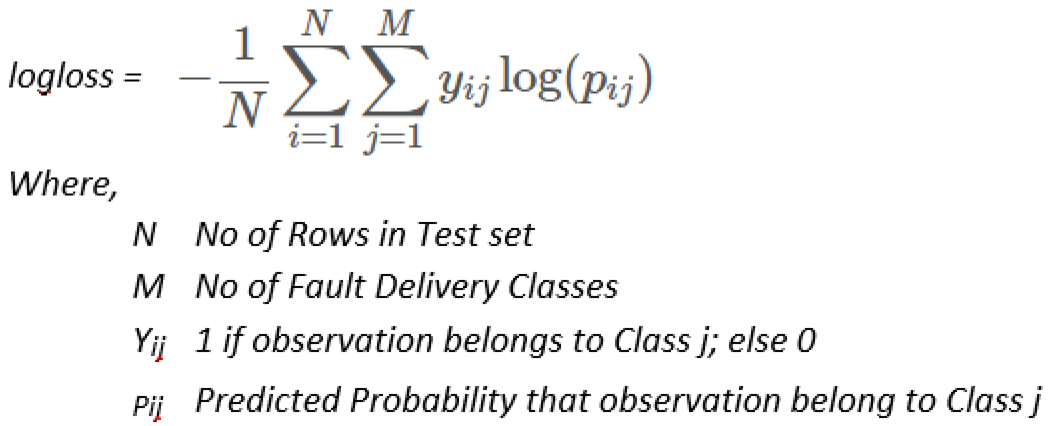
A deep learning algorithm will be developed using Tensorflow/Keras and will be trained with training data. Specifically a CNN will be actualized in Tensorflow/Keras and will be optimised to minimize multi-class logarithmic loss as defined in the Evaluation Metrics section. Predictions will be made on the test data set and will be evaluated.

Benchmark model

The benchmark demonstrated was a simple CNN model with the Public Leaderboard score(multi-class logarithmic loss) of 0.08690. Attempt will be made so that score(multi-class logarithmic loss) obtained will be among the top 50% of the Public Leaderboard submissions.

Metrics

Submissions are evaluated using the multi-class logarithmic loss. Each image has been labeled with one true class. For each image, you must submit a set of predicted probabilities (one for every image). The formula is then,



$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

Where,

N	*No of Rows in Test set*
M	*No of Fault Delivery Classes*
Y_{ij}	*1 if observation belongs to Class j; else 0*
p_{ij}	*Predicted Probability that observation belong to Class j*

The submitted probabilities for a given image are not required to sum to one because they are rescaled prior to being scored (each row is divided by the row sum). In order to avoid the extremes of the log function, predicted probabilities are replaced with max(min(p, 1 − 10−15), 10−15).

The logarithmic loss measures the performance of a classification model by taking the prediction input as a probability value between 0 and 1 rather than a simple true or false. That means the logarithmic loss takes into account the uncertainty of the model prediction based on how much it varies from the actual label, yielding a more nuanced evaluation of the model’s performance.

Project Design

From the description and problem statement it can be inferred that computer vision can be used to utilize to land a solution.

- ✓ Initially data exploration will be carried out to understand possible labels, range of values for the image data and order of labels.
- ✓ This will help preprocess the data and can end up with better predictions.
- ✓ This necessary preprocess functions will be implemented , data will be randomised and Convolution neural networks(CNN) will be implemented in Tensorflow/Keras.
- ✓ Finally necessary predictions on the test data will be carried out and these will be evaluated.