Machine Learning Engineer Nanodegree

Capstone Project Proposal
Distracted Driver Detection

Dhandu Swetha

14th February , 2019

Proposal:

Domain Background:

According to the CDC motor vehicle safety division, there are 3,287 deaths each day due to fatal car crashes. On average, 9 of these daily fatalities are related to distracted driving.

1. Distracted driving accounts for approximately **25%** of all motor vehicle crash fatalities.
2. At the time of fatal crashes, teens have been the largest age group that reported being distracted while driving.
3. Driver distraction is reported to be responsible for more than **58%** of teen crashes.
4. In 2015, **391,000** injuries were caused in distracted driving related accidents.
5. In that same year, distracted driving was cited as a major factor in **3,477** traffic deaths.

State Farm hopes to improve these alarming statistics, and better insure their customers, by testing whether dashboard cameras can automatically detect drivers engaging in distracted behaviours.

Problem Statement:

Given a dataset of 2D dashboard camera images, an algorithm needs to be developed to classify each driver's behaviour and determine if they are driving attentively, wearing their seatbelt, or taking a selfie with their friends in the back seat, talking with the people inside the vehicle etc..? This can then be used to automatically detect drivers engaging in distracted behaviours from dashboard cameras.

Datasets and Inputs:

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

Some of the examples are:

The 10 classes to predict are:

Class	Description
c0	Safe driving.
c1	Texting (right hand).
c2	Talking on the phone (right hand).
c3	Texting (left hand).
c4	Talking on the phone (left hand).
c5	Operating the radio.
c6	Drinking.
c7	Reaching behind.
c8	Hair and makeup.
c9	Talking to passenger(s).

Following are the file descriptions and URL's from which the data can be obtained:

- imgs.zip-zipped folder of all (train/test) images.
- sample_submission.csv - a sample submission file in the correct format
- driver_imgs_list.csv - a list of training images, their subject (driver) id, and class id

▪ https://www.kaggle.com/c/state-farm-distracted-driver-detection/data

Solution Statement:

Our proposed solution consists of a Convolutional Neural Networks(CNN). The convolutional neural networks train on raw images, face images, hands images. A deep learning algorithm will be developed using Tensorflow/Keras and will be trained with training data. Specifically a CNN will be implemented 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 model for this project 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 is the number of images in the test set
- M is the number of image class labels
- log is the natural logarithm, y_{ij} is 1 if observation i belongs to class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs 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 arrive at a solution. CNN class of deep learning algorithm can be employed for this problem. 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. After this necessary preprocess functions will be implemented , data will be randomised and CNN will be implemented in Tensorflow/Keras. Finally necessary predictions on the test data will be carried out and these will be evaluated which means the logarithmic loss takes into account the uncertainty of the model prediction based on how much it varies from the actual label.