Machine Learning Engineer Nanodegree

Capstone Proposal

Sonam Chawla
March 29th, 2019

Proposal (Accident Prevention)

Domain Background

An accident is an unplanned event that results in injury, deaths, damage to property or some other losses. Accidental Deaths can be classified as deaths due to forces/factors of the nature termed as ‘Natural Accidental Deaths’ or due to deliberate or negligent conduct of human beings, ‘Un-natural Accidental Deaths’ or due to causes not covered in the above two categories and with no initial apparent cause of deaths categorised as ‘Other Causes of Deaths’ like poisoning, sudden death etc.

We've all been there: a light turns green and the car in front of you doesn't budge. Or, a previously unremarkable vehicle suddenly slows and starts swerving from side-to-side. When you pass the offending driver, what do you expect to see? You certainly aren't surprised when you spot a driver who is texting, seemingly enraptured by social media, or in a lively hand-held conversation on their phone.

According to National Crime Records Bureau (NCRB), traffic accidents are around 53% of total accidental deaths. According to accidental death report digitised by NCRB in 2015 (http://ncrb.gov.in/StatPublications/ADSI/ADSI2015/chapter-1%20accidents.pdf), there has been 4.9% increase in traffic accidents from year 2014 to 2015 as shown in table below:

Cause	2014			2015			% Variation during 2015 over 2014
	No.	% Share	Rate (No. of deaths per 1,00,000 population)	No.	% Share	Rate (No. of deaths per 1,00,000 population)	4.9
Traffic	1,69,107	53.4	13.6	177423	52.8	13.4	

According to https://www.prsindia.org/policy/vital-stats/overview-road-accidents-india 18-34 year olds are most affected by these traffic accidents. This age group is the most productive group for any country. Losing them can affect the growth of country. This is the reason I was motivated to do this project to help prevent dome accidents.

Problem Statement

Given a dataset of dashboard camera images, an algorithm need to be developed which can estimate if any driver is not driving properly and can lead to an accident. Algorithm should be able to detect driver's behaviour like are they driving attentively, wearing their seatbelt, or taking a selfie with their friends in the backseat?

Datasets and Inputs

Driver images taken in a car with a driver doing something in the car (texting, eating, talking on the phone, makeup, reaching behind, etc) are taken as input. The goal is to predict the likelihood of what the driver is doing in each picture.

Dataset Description

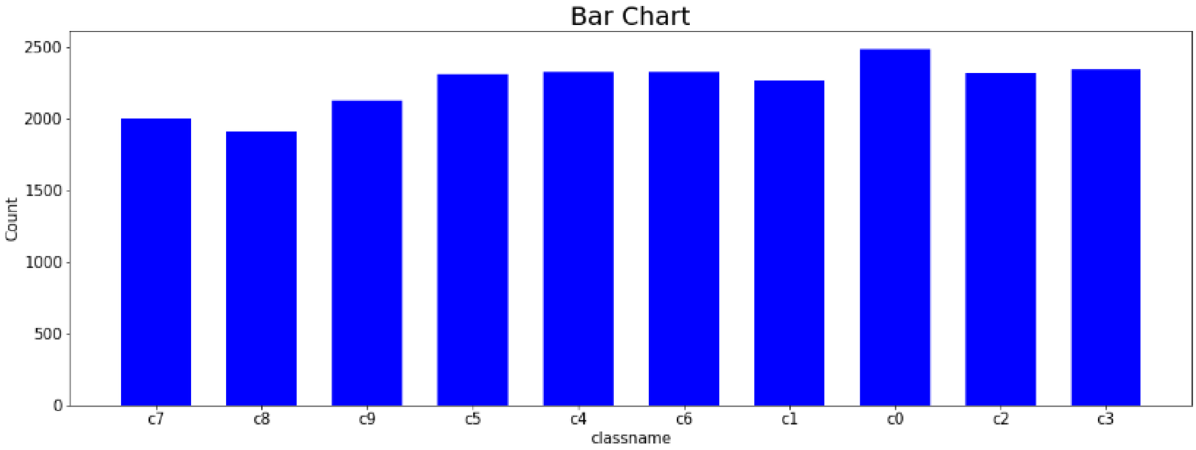
- There are 102150 total images.
- There are 17939 training images.
- There are 10 total training categories.
- There are 4485 validation images.
- There are 79726 test images.

All images are of the same size (640x480). I will resize the image to square dimensions, most probably 224x224.

The 10 classes to predict are:

0. Safe driving
1. Texting – right
2. Talking on the phone – right
3. Texting – left
4. Talking on the phone – left
5. Operating on radio
6. Drinking
7. Reaching behind
8. Hair and Makeup
9. Talking to Passenger

Above classes are evenly distributed among images. All classes have 2000 to 2500 images. Bar Chart displaying the distribution is shown below:



To ensure that this is a computer vision problem, metadata has been removed from data such as creation dates. The train and test data are split on the drivers, such that one driver can only appear on either train or test set.

Following files have been received from Kaggle State Farm Distracted Driver Detection Competition (https://www.kaggle.com/c/state-farm-distracted-driver-detection/data)

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

Solution Statement

Will implement a CNN using Tensorflow/Keras to detect the behaviour of driver in images. Will try to optimize the CNN using multi-class logarithmic loss. Prediction score will be calculated and validated on test data.

Benchmark Model

I will run the linear regression classifier to get the base MSE. Then I will compare my CNN model with it to check my model performs better than linear regression classifier.

Evaluation Metrics

Submissions are evaluated using the multi-class logarithmic loss. Log Loss takes into account the uncertainty of your prediction based on how much it varies from the actual label. For each image, you must submit a set of predicted probabilities (one for every image). The formula is then,

$$logloss=-1N\sum i=1N\sum j=1Myijlog(p_{ij}),$$

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)*.

Project Design

Computer vision will be used to obtain the solution. CNN will be used for the implementation. Tensorflow/Keras libraries will be used to implement CNN.

Initially data exploration will be carried out to understand possible labels, range of values for the image data and order of labels. Then pre-processing of data will be done based on data exploration result. Data randomization will be done to get better predictions. RGB images will be converted to 4D tensor. Will implement Conv2D layers with “relu” activation for CNN implementation. Will optimize using rmsprop optimizer.

Finally predictions on test data will be done and evaluation will be done on same.