

# MLND Project Proposal

September 3, 2018

## 1 Machine Learning Engineer Nanodegree

### 1.1 Capstone Proposal

Michael Stevens  
September 2nd, 2018

### 1.2 Proposal

#### 1.2.1 Domain Background

The StateFarm Kaggle Competition represents an important application of machine learning classification to image data. Distracted driving is a large contributor to property damage and personal injury. If it is possible to add a camera facing the operator of a motorized vehicle, it may be possible to recognize when the operator is distracted. This could provide an opportunity to add a safety system similar to the audible chime when the seatbelts are not in use. The identification of a distracted operator could be used to remind the operator, alert other drivers with a signal, alter insurance billing, or be used as an input into a self-driving auto-brake system.

This problem appears to be feasible to solve using the skills taught in the Udacity MLND. This problem has not only been the subject of the Kaggle competition, but also the subject of academic projects. Other research has shown the ability of deep neural networks to perform pose estimation on subjects. These projects show that this task is possible to learn using machine learning.

- Kaggle StateFarm Distracted Driver Detection
- <https://www.kaggle.com/c/state-farm-distracted-driver-detection>
- End-to-End Deep Learning for Driver Distraction Recognition (Koesdwiady)
- [https://www.springer.com/cda/content/document/cda\\_downloaddocument/9783319598758-c2.pdf?SGWID=0-0-45-1608335-p180889205](https://www.springer.com/cda/content/document/cda_downloaddocument/9783319598758-c2.pdf?SGWID=0-0-45-1608335-p180889205)
- DarNet: A Deep Learning Solution for Distracted Driving Detection
- [https://users.cs.duke.edu/~cdstreif/static/darnet\\_presentation.pdf](https://users.cs.duke.edu/~cdstreif/static/darnet_presentation.pdf)
- DeepPose: Human Pose Estimation via Deep Neural Networks (Toshev)
- <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/42237.pdf>

### 1.2.2 Problem Statement

The problem is to classify images of automobile operators into one of ten possible classes. One of the classes represents normal driving and the other nine represent distracted driving. The classification algorithm to solve this problem will output probabilities for each possible class. The performance of the classification will be measured using multi-class logarithmic loss.

This classification problem appears to have several distracted classes that should be possible to classify with CNN deep learning systems, such as using a cell phone or reaching into the back seat. A classifier that generalizes to a test set could be useful in a larger integrated automobile safety system.

The full list of classes is the following. \* 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

### 1.2.3 Datasets and Inputs

The dataset for this problem was acquired by StateFarm. The drivers captured represent a mix of genders, clothing, eyewear, ethnicity, and weights. Several vehicles are represented in the training dataset. A number of the images show the drivers wearing StateFarm badges, so they may have used their employees for a majority of their captures. The subjects captured were not actually operating the vehicle because the car was being towed behind a truck. In some images, a person with a clipboard is visible in the back seat. Each labeled class for has between 1911 and 2489 images.

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

The training dataset is organized into ten folders, with one representing each class. Each image is a 640x480 pixel jpeg captured from a location near the passenger side A-pillar. A CSV file provides a driver identifier, class, and image name for each image in the training folder. A separate folder called test has 79,726 images, but no labels are provided as a part of the download, so they will not be used. Instead, the images in the test folder will be divided into train and test sets by splitting on the driver id.

### 1.2.4 Solution Statement

A solution to this problem would train a deep CNN network using the training data and labels. Deep CNNs have been applied successfully to many classification tasks, such as recognizing the breeds of dogs and pose estimation. To solve this problem using a deep neural network can be trained on a subset of the 22,424 labeled images. With the smaller number of labeled images available, it is likely to require transfer learning on a pretrained network.

### 1.2.5 Benchmark Model

A benchmark model for distracted driving would be to always predict normal undistracted driving with complete certainty, 1.0, and all distracted driving classes as 0.0. The predictions will then be measured with multi-class logarithmic loss. Beating this benchmark's log loss is a bare minimum. The Kaggle competition also provides other benchmarks using the leaderboard with the log loss of the winning entries being lower than 0.10.

### 1.2.6 Evaluation Metrics

A standard metric for evaluation of multi-class classifiers is the log loss metric. This is what the Kaggle competition originally used and is definitely suitable. Log loss takes into account the certainty result for the prediction. A standard accuracy measurement only accounts if the top prediction matched the actual label. In the case of a 10-class classification problem the actual class could be predicted with 0.1 + epsilon certainty or 1.0 certainty, but both would be counted the same by an accuracy score. Log-loss would rate the 1.0 certainty prediction much better than the 0.1 + epsilon prediction.

Log loss heavily penalizes mispredictions with a low predicted probability with extremely high loss values. Correct predictions with absolute certainty score close to zero. The undefined values of the formula for zero and one certainties are compensated for by applying a minimum and maximum to ensure the p values are within (0+epsilon, 1-epsilon).

- [http://wiki.fast.ai/index.php/Log\\_Loss](http://wiki.fast.ai/index.php/Log_Loss)
- [http://scikit-learn.org/stable/modules/generated/sklearn.metrics.log\\_loss.html#sklearn.metrics.log\\_loss](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html#sklearn.metrics.log_loss)

### 1.2.7 Project Design

For initial data analysis it will be useful to view some grids of images showing both all classes for a single person. Additionally, looking at a grid of images representative of each class will be useful. It is possible that some interesting properties of these images could be discovered or some mislabelings could be present.

The first steps to using this dataset with transfer learning would be to resize the images to a compatible input size of the neural network being used for transfer learning, such as one of the ResNet variants. Getting even a few passes of training will show the full pipeline is working. For simplicity, the reduced input size will be used with the from scratch network.

Attempting a deep neural network from scratch will require paying careful attention to the potential for vanishing gradients and the small amount of training data available. Some usage of recurrent layers may be necessary to deal with the vanishing gradient problem. Dropout layers may also be necessary to ensure the classification generalizes to the test set and doesn't start to memorize the training data. The relu activation function will be used as it has better gradient propagation than tanh or sigmoid.

For both the from scratch neural network and the transfer learning, the networks will be trained on the full training set using k-fold cross validation until the network shows signs of overfitting or is no longer increasing in predictive power on the k-fold cross validation sets. This will be done by plotting learning curves. When the training and cross validation scores are appear to be converging, the training will be complete. It is unlikely that the full network architecture of ResNet or something similar could be trained with the limited amount of data.

To understand the classification, a sampling of the mispredictions will be examined manually to see if there is any obvious pattern to how the misclassification occurs. An analysis of the classes that are most frequently mispredicted could also be useful in debugging the classifier.