Machine Learning Engineer Nanodegree

Capstone Proposal

Rohan Patel
March 20, 2018

Proposal

Domain Background

Deep Learning is a hot topic in machine learning and an area of interest of mine. Deep Learning and neural networks have many use cases, but one area I am particularly interested in exploring is image classification and computer vision. Image classification is a type of learning problem in which a trained model must assign an image a category. These categories can range from actions, objects, letters, numbers, etc. One possible use case for image classification is determining if a driver is distracted at the wheel based on images taken from dashboard cameras. Thousands of people are injured and hundreds are killed each year by distracted drivers. Determining if a driver is prone to distraction can help cars better alert drivers as well as provide valuable information to insurance companies.

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

Problem Statement

The CDC motor vehicle safety division states that one out of five car accidents are caused by distracted drivers. The insurance and financial services company, State Farm, is hosting a competition for people to come up with ways to classify a driver's behavior based on 2D dashboard camera images. The goal of this competition is to allow State Farm to help improve the above statistic as well as better insure their customers, by determining whether it is possible to determine bad driving behavior from dashboard cameras.

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

Datasets and Inputs

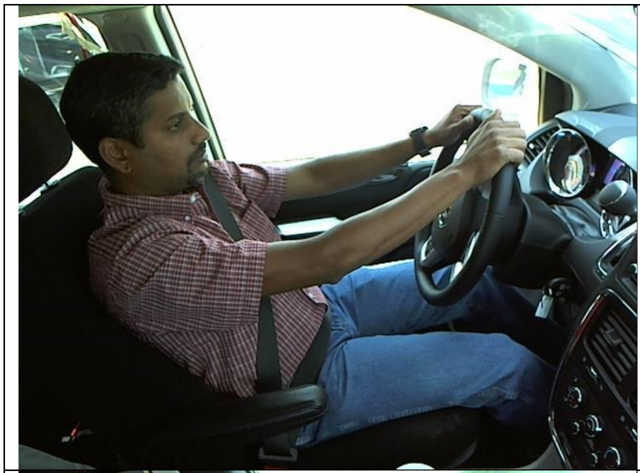
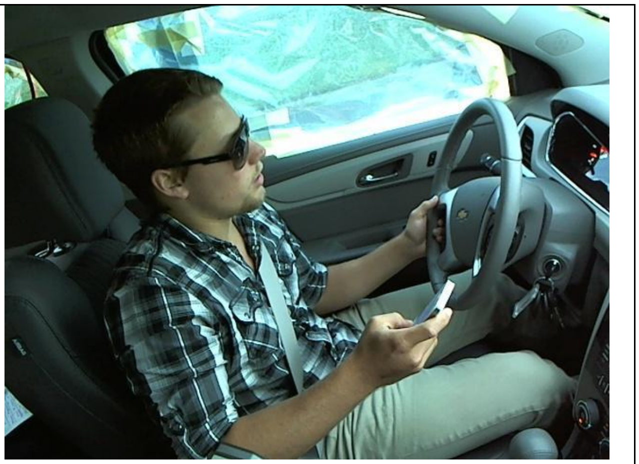
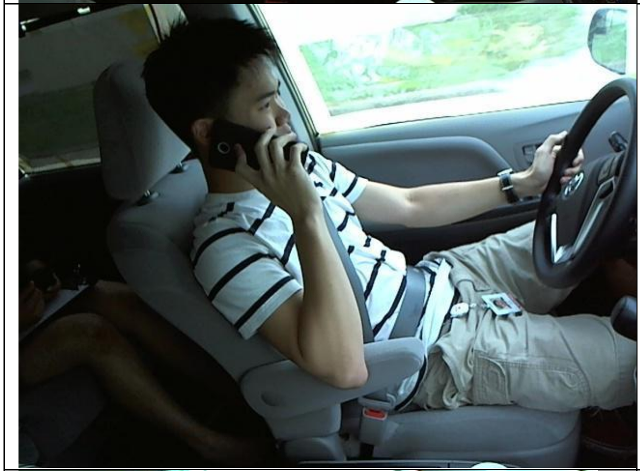
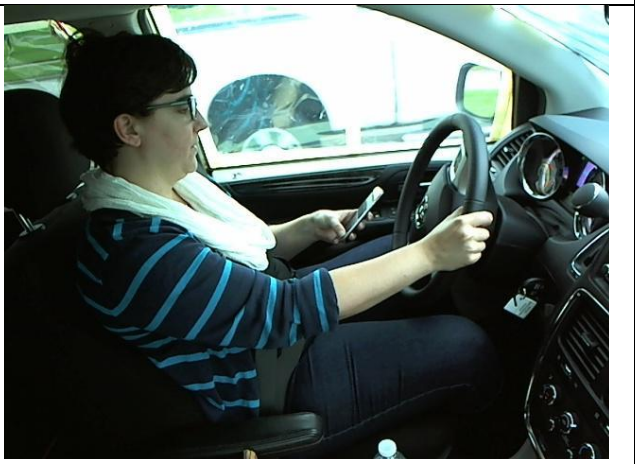
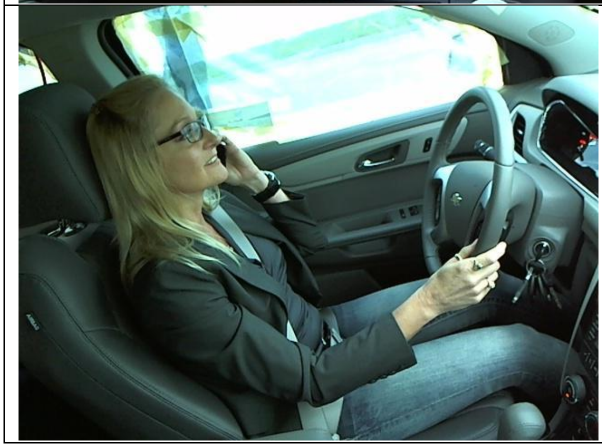
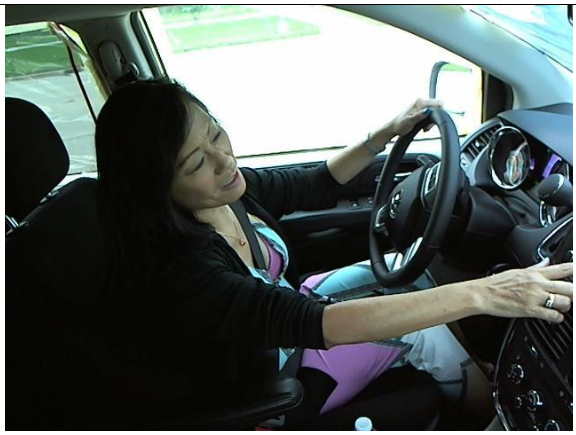
The competition has provided a large dataset of about 100 thousand color images for training/testing data. These are 2D dashboard camera images of drivers either distracted or safely driving. In addition, this is a labeled dataset with one of the 10 following classifications.

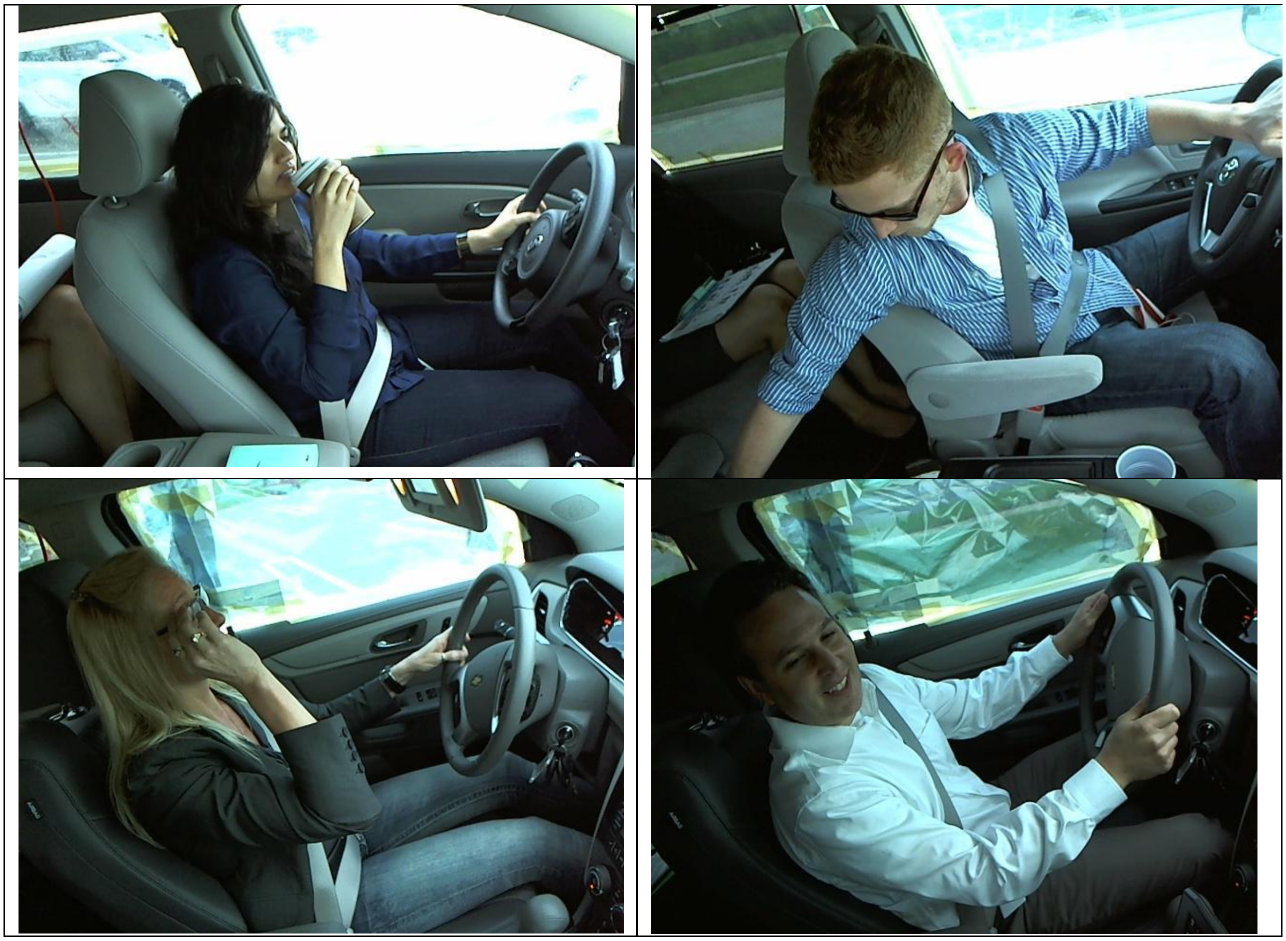
- c0: safe driving
- c1: texting - right
- c2: talking on the phone - right
- c3: texting - left
- c4: talking on the phone - left
- c5: operating the radio
- c6: drinking

- c7: reaching behind
- c8: hair and makeup
- c9: talking to passenger

The supplied driver images are taken in a car while the driver is performing one of the actions above. As the problem statement stated, the goal of this project is to determine/predict the driver's action in a given image.

Here are a few sample images that show the actions c0 to c9 from left to right:



The images provided have removed all metadata such as creation dates and other marks. In addition, the training and test data are split on the drivers so that drivers can only appear on either train or test. In other words, if they do appear multiple times they will appear on the same set of data. This is good because we want to make sure that our testing data and training are independent and not cross contaminated.

In total the data consists of the following:

- 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 supplied images from the training set will be used to train a classifier which will then be tested on the given training set.

In addition, we may want to preprocess the data so that they are all the same size. Also, color may also be a factor. We may choose to convert all the images to grey scale and provide as input a 2D array of values corresponding to grey scale pixel values. On the other hand, we may choose a 3D array input where we separate the RGB channels.

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

Solution Statement

To solve this problem, we must first understand what we are trying to achieve. We know that this is a labeled data set, thus it is a form of supervised learning. We also know that we are working with 2D images and want to classify them. In my opinion, the best course of action would be to use a Convolutional Neural Network. These are great at classifying images. We may choose to create our own architecture or use transfer learning to use a subset of a previously trained classifier.

Benchmark Model

The Kaggle website provides a leaderboard for scores of previous submissions obtained using the evaluation metric (multi-class logarithmic loss) described below. The goal of this capstone is to place as high on the leaderboard as possible by minimizing the log loss function. As a standard for this project a log loss of 3.0 is the minimum acceptable to place in the top 1400. However, the real goal and challenge is to place much higher.

Evaluation Metrics

The model will be evaluated on the multi-class logarithmic loss function. Remember that each image has been labeled with the driver's behavior from a category c0 – c9. The Convolutional Neural Network will supply a vector of probabilities for each of these categories for each supplied image:

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

N = number of images in the test set
M = number of image class labels
Log = natural logarithm
Yij = 1 if observation i belongs to class j and 0 otherwise
Pij = predicted probability that observation i belongs to class j

The predicted probabilities for each given image are not required to sum to one because are rescaled prior to being scored (each row is divided by the row sum). Also, to avoid extremes given by the log function, the predicted probabilities are replaced with the following:

$$max(min(p, 1-10^{-15}), 10^{-15}).$$

https://www.kaggle.com/c/state-farm-distracted-driver-detection#evaluation

Project Design

The steps needed to complete this project are fairly straight forward, but obstacles will need to be overcome without a doubt. In essence, the main steps include environment setup, data collection, data importing, data preprocessing, creating train/test sets, choosing the machine learning model, training the model, evaluating the model, parameter tuning, and testing.

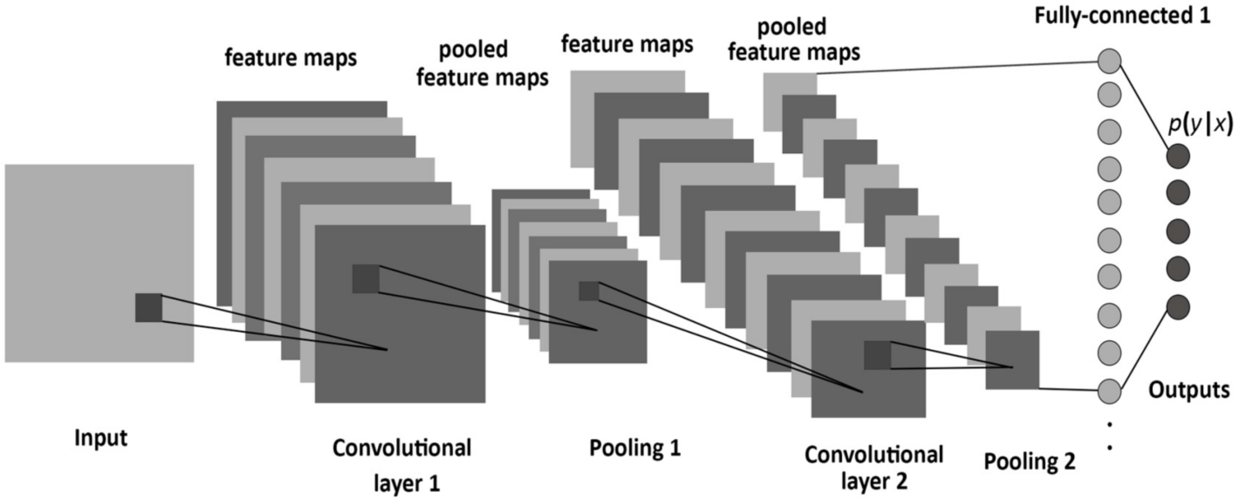
In the environment setup stage we will create an environment to create and test our machine learning models. We will be using tensor flow and more specifically tensorflow-gpu to locally train our models using a graphics card.

The data collection portion has already been completed by kaggle and they have been kind enough to provide us over 100K driver images. These images have been labeled as well to help with this supervised learning problem.

The data preprocessing portion will involve image scaling and maybe coloring. We would want to rescale all our images to the same size if they haven't been already. In addition, we will need to decide whether we want to work with grey scale images in a 2D array format or color images in a 3D format with the third dimension being color channels.

Luckily for us, the data has already been split in training and testing sets by kaggle. We have a very large training set of roughly 20K images and an even larger testing set of 80K images.

Model selection will be one of the challenging factors in this project. We will be using a convolutional neural network. We will either design one from scratch or use transfer learning to use a subset of a previously trained model. However, both will output a layer with 10 nodes that correspond to the 10 classifications for drivers (c0 – c9). Here is a sample image of a CNN:



Training the model, evaluating the model, and tuning hyper parameters will be done several times to minimize the training error. We will possibly go through many different convolutional neural network architectures and change several parameters to find the correct formula. We will also, test our model on the testing set to see the testing error. This will allow us to see our model's performance as well recognize instances of overfitting or underfitting.

The last step in this project will be to check our ranking against our benchmark which is the kaggle leaderboard. We will try to keep improving our log loss function until we reach a satisfactory value.

References

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

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

https://www.kaggle.com/c/state-farm-distracted-driver-detection#evaluation