Capstone Milestone Report

Project: Distracted Driver Detection (Kaggle Competition)

## Manasa Raghavan

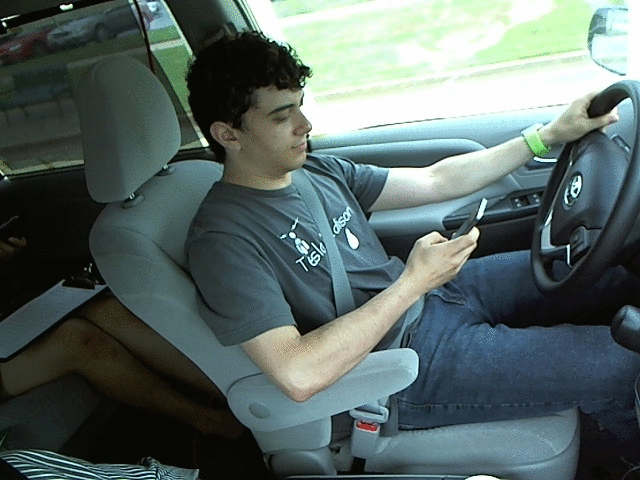
## Mentor: Ankit Jain

Motivation:

**Can Computer Vision spot a distracted Driver?**

According to the CDC motor vehicle safety division, [one in five car accidents](http://www.cdc.gov/motorvehiclesafety/distracted_driving/) is caused by a distracted driver. Sadly, this translates to 425,000 people injured and 3,000 people killed by distracted driving every year.

State Farm has sponsored a competition to detect a distracted driver using 2D image captures from a dashboard camera.



With a simple dashbaord camera if we can detect and deter a distracted driver, we could potentially avoid a number of accidents.

Initial Data Exploration:

2D image captures from a dashboard camera are categorized as

* 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

****

Training Data:

The training data for each class is in a sub-directory labeled with the class (c0-c9). The images in the training data is from 26 different subjects. The input file driver\_imgs\_list.csv provides the mapping between driver ID and image ID. This is useful for splitting data based on the driver ID into cross-validation sets or test sets for accuracy prediction. The images are of 480x680 pixels resolution with 3 color channels.

Test Data:

The test data contains 79K images. Few selected images below that can be easily mis-classified or ambiguous:

Correct: c8: - hair and makeup

Incorrect: c1: texting - right



Correct: c0: safe driving

Incorrect: c9: talking to passenger

# C:\Users\Vibhav\Documents\Kaggle\Driver distraction\input\test\img_100721.jpg



Problem Statement:

The Distracted Driver detection is an Image classification problem. The goal is to predict the likelihood of what the driver is doing in each picture.

Image Classification Tools:

Deep Neural Networks have led to breakthrough results in computer vision problems using pattern recognition. Convolutional Neural Networks (CNN) is a special kind of Neural network is used in deep learning. The popularity of using CNNs for image processing can be widely attributed to the rise of GPUs for fast computation.

**A brief description on CNN:**

CNNs have a structure that can enable feature extraction from images. CNNs are different from fully connected Neural networks in using convolved features as opposed to the entire pixel array in the layers. Using convolved features helps with images where the features in a region are sparse and also that features or statistics in one part of an image are similar to other parts.



Figure Example of Convolved Feature

Another technique used is called pooling. Pooling downsamples the features further and uses aggregate statistics of the features for classification. Pooling helps avoid over-fitting and also provides translation invariance.

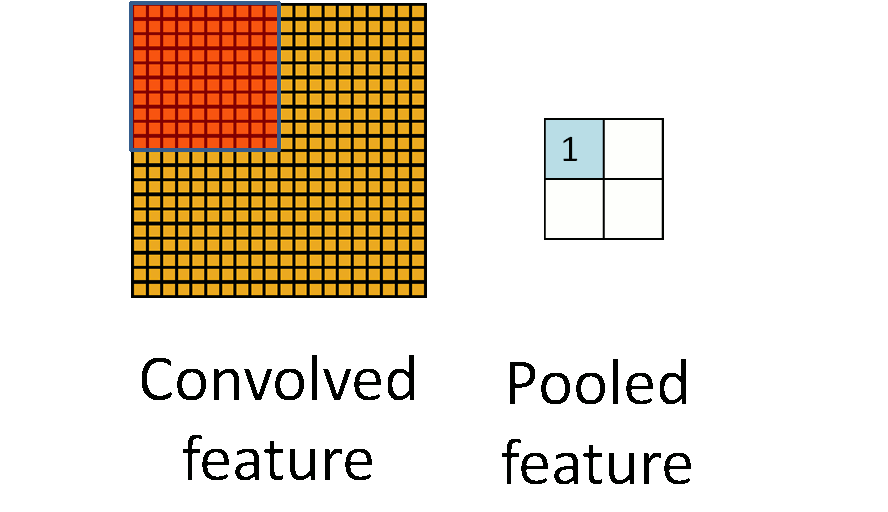


Figure : Pooling from Convolved Features

A combination of convolution and pooling (sub-sampling) layers are used for feature extraction before using a fully connected Neural network for classification.

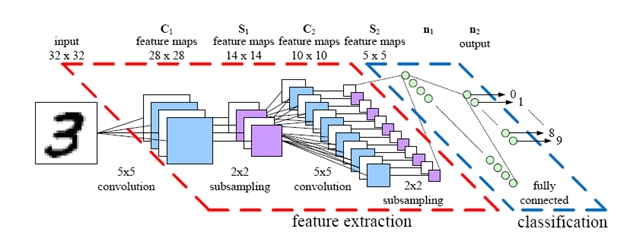


Figure : Typical CNN Architecture for Image classification

**Implementing CNN:**

Several packages are available in Python for implementing CNN - Caffe, Tensorflow, Theano, Keras, etc. Tensorflow isn't very easy to install and work with on a Windows machine. It needs a Docker/ virtual machine to run in Linux environment. After some initial struggle with getting it to work with other libraries like OpenCV, I opted to use Keras with Theano backend.

Progress with Classification:

To work without a GPU for image classification, the images need to be resized for faster computation. From 480x680 RGB, the images were resized to 72x102 and grayscale. The cross-validation set was created using images from a driver not in the training set. With the initial setup for the CNN using 8 layers, Convolution kernel of 2x2, and pooling size of 2x2, the measured accuracy on the CV set with 10 fold cross-validation was around 15%. The final result from the test data was submitted on Kaggle and the score obtained was 2.05809 on the public leader board.

Work ahead:

Fine tune the parameters of the CNN. Experiment with different convolution kernel, pooling sizes and drop out values.