Real time detection of driver distraction using convolutional neural network

***Abstract:*** *Distracted driving is the main cause for large number of motor vehicle accidents in all over the world. Detecting the distracted driver is considered to be major potential area of research for reducing the road accidents. This paper focus on a methodology to reduce the accidents caused by distracted driver by using the deep neural networks. A convolutional neural network is used to develop the actions of driver from driver image data set and this information is used to classify the distracted driver. Proposed system consist of 3 Models, vanilla CNN model, vanilla CNN with data Augmentation and CNN with transfer learning. Deep neural network model is developed from state farm dataset which consist of 10 actions of 26 different subjects such as texting, operating the radio, talking on phone, reaching behind, normal driving, drinking etc. Results obtained from 5 Epochs shows that all experiments exceeded 90% accuracy and best result was found to be 97%*

*NOVELTY*

*The project aims at building a model which will detect the distracted drivers with high accuracy. So I experimented with 3 different model which use 3 different approach to solve the problem.*

1. *Improved vanilla CNN model is an improved version of normal CNN model which uses 4 convolutional layer, one flatten layer, and 2 dense layer.*

*This model gives an accuracy of 97.7%*

1. *Vanilla CNN model with data augmentation is similar to first model. here we will   generate more images using****ImageDataGenerator****and split the training data into 80% train and 20% validation split.it has a total of 3 convolutional layer, a flatten layer, and 3 dense layer. This model produce an accuracy rate of 97.1%.*

## *CNN with Transfer Learning (VGG, MobileNet) reduce training time with much compromising accuracy. This model will produce 76% accuracy in very less time.*

In this project, I'll use the dataset which includes images of drivers while performing a number of tasks including drinking, texting etc. The aim is to correctly identify if the driver is distracted from driving. We might also like to check what activity the person is performing.

The notebook will be broken into the following steps:

1. Import the Libraries.
2. Import the Datasets.
3. Create a vanilla CNN model.
4. Create a vanilla CNN model with data augmentation.
5. Train a CNN with Transfer Learning (VGG16).
6. Experimental Results.

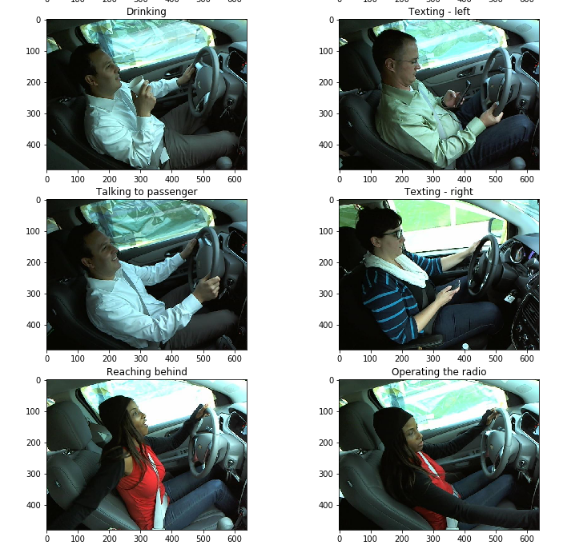
**Import the Libraries**

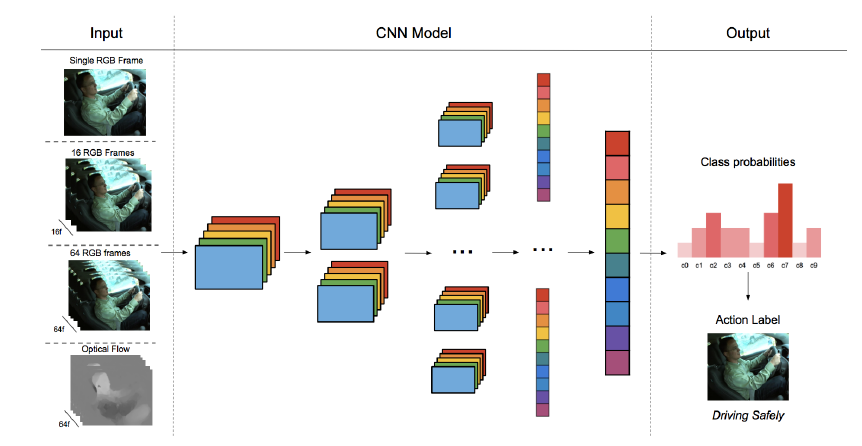
I'll use Keras and Tensor flow libraries to create a **Convolutional Neural Network**. So, I'll import the necessary libraries to do the same. The model is built using Google colab research platform, which provides high performing setup to train and test the model.

**DATASETS**

We use State Farm Dataset for training and testing the model. State farm, an insurance company released dataset which contains driver images for Kaggle competition in 2016 for image based driver poster classification. The dataset consist of 22,450 labelled images of 26 subjects which includes different colour, ethnicity, action, age, size etc. these subjects were used to perform classification of the subjects into 10 classes such as normal/safe driving, talking in the phone by both left and right hands, controlling the radio, text messaging by left hand, test messaging by right hand, talking to the co passenger, drinking while driving, hair and makeup, reaching behind, etc. each image is labelled with their action class.



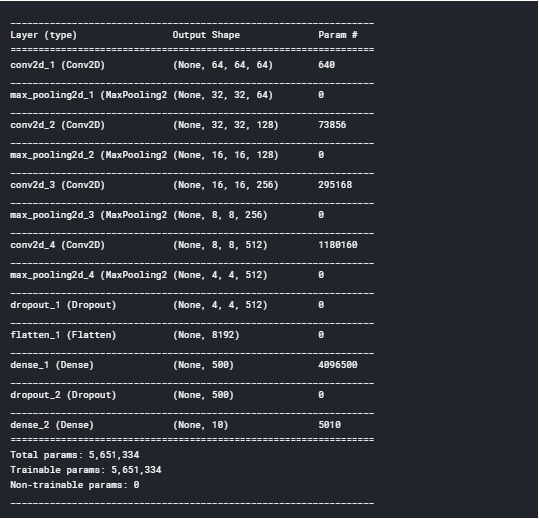


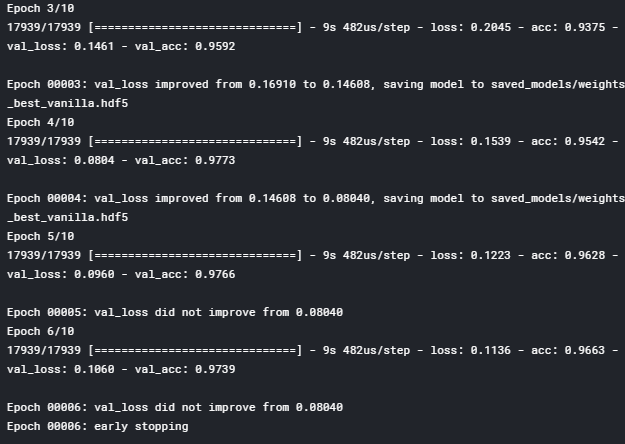


**MODEL 1**

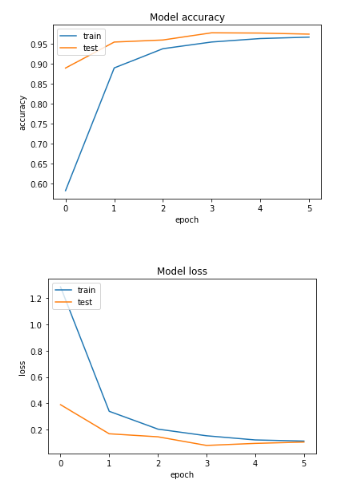
**IMPROVED VANILLA CNN MODEL**

This model is an improved version of normal CNN model which uses 4 convolutional layer, one flatten layer, and 2 dense layer.  I'll use the optimizer as rmsprop, and loss as categorical\_crossentropy. Activation function for 4 convolutional layers are ReLU and activation function for the first dense layer is ReLU and the last dense layer is softmax.





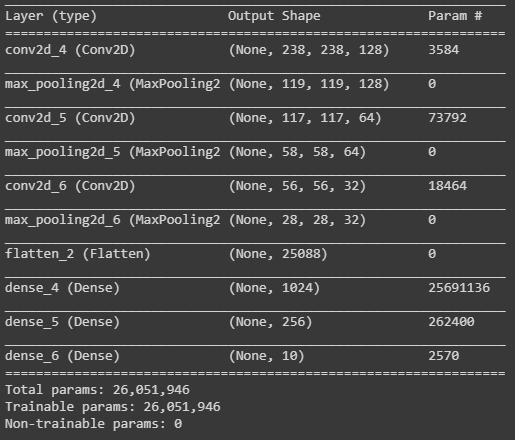
This model gives an accuracy of 97.66%



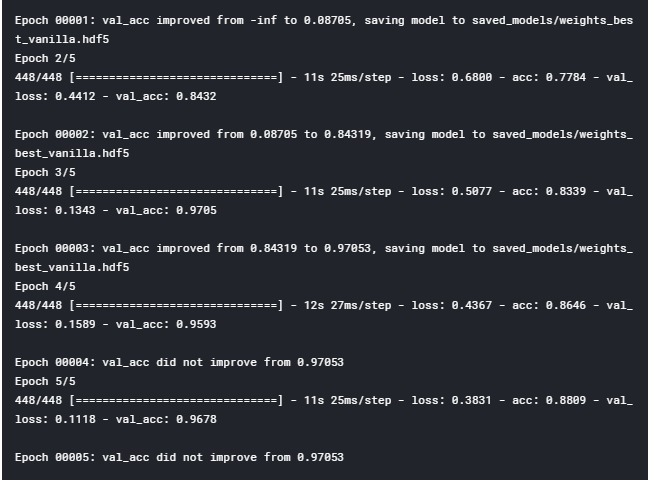
**MODEL 2**

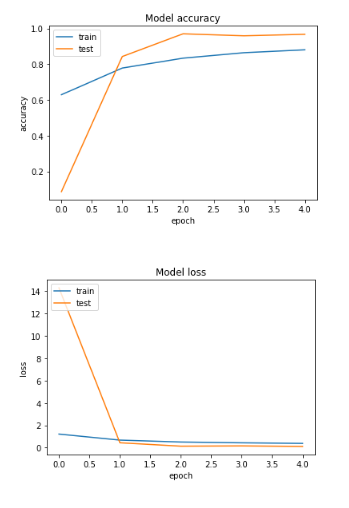
## Vanilla CNN model with data augmentation

## Here I'm augmenting the previous model classifier, I'll use the data on which I want to train the model. The folder train includes the images I need. I'll generate more images using ****ImageDataGenerator**** and split the training data into 80% train and 20% validation split. Vanilla CNN model with data augmentation is constructed with a total of 3 convolutional layers, then a flatten layer and 3 dense layer. We use *Adam* optimizer and *categorical\_crossentropy* for loss. Activation function for 3 convolutional layers are ReLU and activation function for first two dense layer is ReLU and last dense layer is sigmoid.



*.*





**MODEL 3**

## CNN with Transfer Learning (VGG, MobileNet)

To reduce training time without losing the accuracy, we train CNN using transfer learning. We use ReLU and softmax activation function for this model. Model configuration of CNN



