Machine Learning Engineer Nanodegree

Nitin Meena

December 16,2018

Report

Project Overview

According to the CDC motor vehicle safety division, one in five car accidents 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 hopes to improve these alarming statistics, and better insure their customers, by testing whether dashboard cameras can automatically detect drivers engaging in distracted behaviours.

So,I had created a CNN model to predict what driver is doing giver driver images.

Problem Statement

Given a dataset of 2D dashboard camera images, Create a model to classify each driver's behavior. Are they driving attentively, wearing their seatbelt, or taking a selfie with their friends in the backseat?

To solve the problem build and train a CNN model and test it on the given test dataset.

Metrics

Submissions are evaluated using the multi-class logarithmic loss. Each image has been labeled with one true class. For each image, we must submit a set of predicted probabilities (one for every image). The formula is then,where N is the number of images in the test set, M is the number of image class labels, *log* is the natural logarithm, *yij* is 1 if observation *i* belongs to class *j* and 0 otherwise, and*pij 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 $max(min(p, 1 - 10-15), 10-15)$

Analysis

Data Exploration

In this project, we are given driver images, each taken in a car with a driver doing something in the car (texting, eating, talking on the phone, makeup, reaching behind, etc). **Our goal is to predict the likelihood of what the driver is doing in each picture.**

The 10 classes to predict are:

- c0: safe driving
- c1: texting - right
- c2: talking on the phone - right
- c3: texting - left
- c4: talking on the phone - left

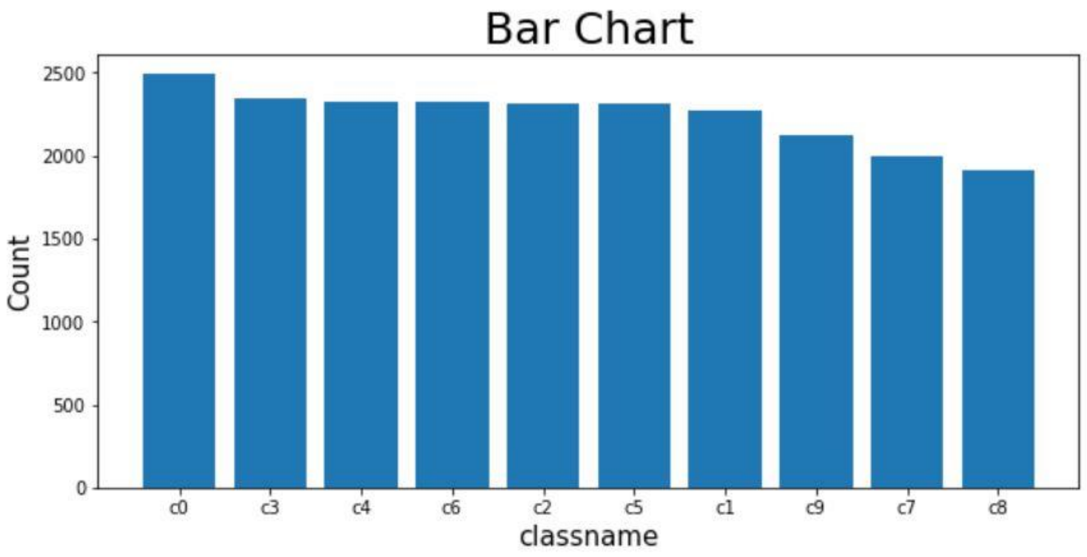
- c5: operating the radio
- c6: drinking
- c7: reaching behind
- c8: hair and makeup
- c9: talking to passenger

File descriptions

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

Exploratory Visualization

From the Graph we can see that no Class Imbalance is there in dataset



Algorithms and Techniques

I have used Convolutional Neural Network and Transfer Learning using Pre-train VGG16 model. In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery.CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing.

Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task. Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned.

Benchmark

The Highest Score on Kaggle is 0.08739.Model should obtain score in top 50% of the Public Leaderboard submissions.

Methodology

Data Preprocessing

Following Preprocessing Steps were taken:

1. Dividing Training Set in Training and Validation set
2. Resizing images to 224 X 224 pixels
3. Normalizing the images by dividing by 255
4. making a extra dimension for batch size

Implementation

In the Simple CNN model,we have created 4 convolutional layers with 4 max pooling layers and batch normalization between. Filters were increased from 16 to 256 in each of the convolutional layers. Also dropout was used along with flattening layer before using the fully connected layer. Relu activation function was used for all other layers .Number of nodes in the last fully connected layer were setup as 10 along with softmax activation function.

Refinement

The CNN model build from scratch performed poorly. It scored a public score of 5.74465. To improve this score another transform model was made using VGG16 model and fully connected layers along with softmax activation function. This VGG16 model fairly well but not up to the expectation. So another VGG16 model was created with fine tuning the top layer containing global average pooling layer and a fully connected layer, where each node for each driver category and is equipped with a softmax.It scored public score of 1.51970

Result

Model Evaluation and Validation

CNN model from Scratch- 5.74465
CNN model with VGG16 bottleneck feature- 2.67432
CNN model with VGG16 with tuning top layers - 1.51970

Justification

To achieve Higher score we need more powerful hardware .To train the CNN model with VGG16 It more then 3 hours and to create bottleneck features it took 6 hours.By having more powerful algorithm(like using VGG19,Res-Net50)and hardware score can be improved

Conclusion

Free-Form Visualisation

VGG16 top layer tuning model performed marvellous.The whole idea of a machine able to tell what driver is doing is great.

Some images Classified by VGG16 models are:

Reflection

Problem I faced during the project are :

1. Pre processing the dataset,Dataset was huge so I had to use H5py library to save the dataset
2. Creating bottleneck Features ,Laptop GPU was not powerful and dataset was too large
3. Training and testing VGG16 tune tuning model,as test data was 3 GB after bottleneck featuring

Improvement

Model can be improved by :

1. Resizing Images to bigger size than 224x224
2. Using VGG19,Res-net 50 or other model
3. Having Powerful GPU so that epoch is more

Reference :

1. **https://github.com/nitml/Dog-Breed-Classifier-**
2. **https://kaggle.com/c/state-farm-distracted-driver-detection**
3. **https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html**
4. **Medium.com**
5. **Tensorflow.org**
6. **Keras.io**