Distracted Driver Detection

Sarath Chandra Reddy (MT19037), Kasarla Mani Kumar Reddy (MT19065), Kolla Nikhil (MT19123)

Introduction

Road Accidents are increasing day by day. Driver Distraction is one of the main causes of road accidents. So, the detection of driver distraction is very important in today's scenario. The aim of our project is to build a model that detects the distraction of drivers in vehicles using various machine learning techniques with the help of dashboard cameras. Input to our model is an image of driver taken in a car while driving and model outputs a predicted type of distraction activity of the driver.

2. Related Work

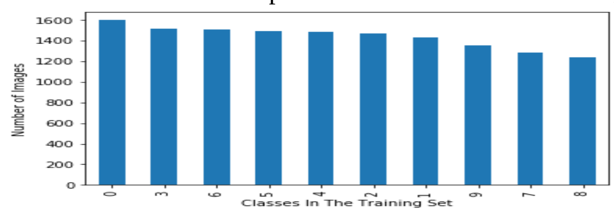
1. Two main topics that researchers focus on are image pre-processing technique and classifying model selection. As for image preprocessing, HaQ et al. proposed that conducting feature extraction instead of directly conducting image flattening might, though not guaranteed, improve prediction accuracy Language.
2. Yuang Liang et al. have conducted study on detecting cognitive distractions in real time they used SVM. He also studied the same using Bayesian networks.
3. A similar study was conducted by Arief koesdwiady et. al. their work uses the framework namely VGG-19 architecture with CNN they have utilized the concept of transfer learning. Features extracted from pre-trained VGG-19 model is state of art frame work.

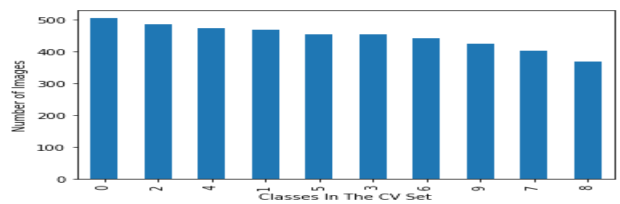
3. Dataset and Evaluation

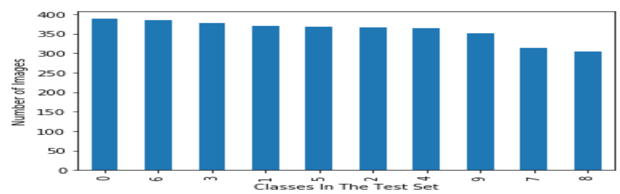
1. The dataset we will be using for this project is from https://www.kaggle.com/c/state-farm-distracteddriver-detection/overview.
2. The Dataset contains 22,424 images categorized into 10 classes (9 classes of distraction and 1 class without distraction or safe driving). Each image size is 640 x 480 pixels. Different classes present in the data set are,

• c0 - Safe driving
• c1 - Texting (right hand)
• c2 - Talking on the phone (right hand)
• c3 - Texting (left hand)
• c4 - Talking on the phone (left hand)
• c5 - Operating the radio
• c6 - Drinking
• c7 - Reaching behind
• c8 - Hair and makeup
• c9 - Talking to passenger(s)

The dataset has been split into Train, CV(held out), Test sets with the count of respective classes in the datasets as,







3.1 Evaluation Metrics
•Accuracy
•ROC
• Multi Class Log Loss
• F1 score

4. Methodology

4.1 Naïve Bayes

1. We have done basic pre-processing of images such as re-sizing (for computational feasibility), Noise elimination, standardization of pixel values with 0 mean and 1 standard deviation, grey scale to RGB channel conversions

(capturing more information from the pixels) of images (if any).

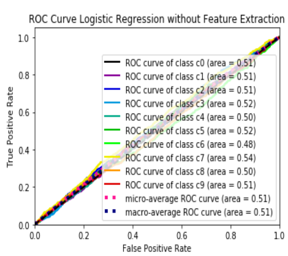
2. We had implemented Naïve Bayes (base-line) model and achieved macro f1-score of 60.3 for training data and macro f1-score of 57.95 for test data.

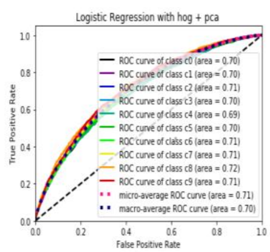
3. The reason for which we had chosen Naïve Bayes as base-line as it does not have any hyper-parameter to tune and will act as good starting point.

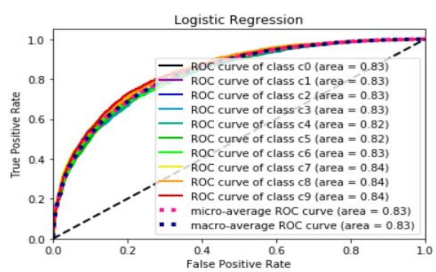
4.2 Logistic Regression

Logistic Regression is the linear model. The output of the model says with what probability the input belongs to that class. When the model was implemented without any feature extraction technique the outputs are random. Accuracy for that initial naïve model was 51. When the model is implemented with histogram of oriented Gradients (hog) along with PCA (dimensionality reduction) got weighted ROC area of 70. After applying grid search cv to tune the hyper parameters of logistic regression the final model got an weighted ROC area of 80 with accuracy around 68.

Plots :

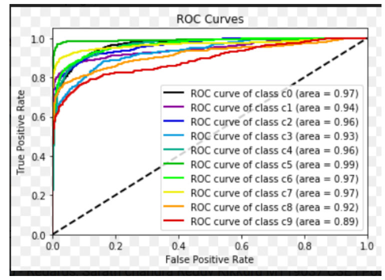






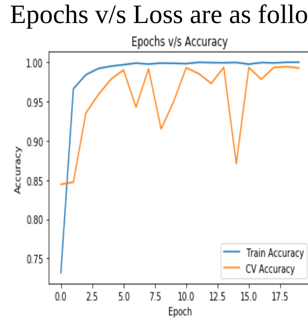
4.3 SVM (Support Vector Machines)

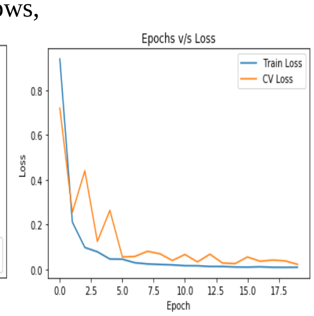
SVM gives the hyperplane that best separates the data. We implemented SVM initially with default hyper parameters. The accuracy is around 55. After tuning of hyper parameters with grid search cv the accuracy is around 76 with weighted ROC area is around 90.



4.4 Neural Networks (Multi-Layered Perceptron)

We have implemented the multi-layered perceptron model for the problem and had built up-on the accuracies by trying out various possibilities. The initial accuracy was around 51.2% when we had built a naïve model with random parameters. Then we added the layer of batch normalization between layers then we found a 10% rise in the accuracy value then we further added the dropout and increased the number of epochs which lead to the further increase of accuracy by around 12% after this due to large fluctuation of accuracies we applied random search technique to obtain the optimal hyper parameters values. The plot of Epochs v/s Accuracy of train and validation and Epochs v/s Loss are as follows,





5. Results and Analysis

5.1 Naïve Bayes

Since, Naïve Bayes is a linear classifier its discriminative power is less compared to non-linear classifiers like kernelized SVM or Neural Networks. Another reason for poor performance of Naïve Bayes might be as it uses conditional probability on assuming conditional independence of features which is not true in case of images because pixels are co-related with each other.

Macro F1-Score – 57.95 on test data.

5.2 Logistic Regression

Since it is a linear model its predictive power is relatively less. In terms of model complexity, it will have high bias and low variance so it will produce less accuracies with respect to high dimensional feature space like images data

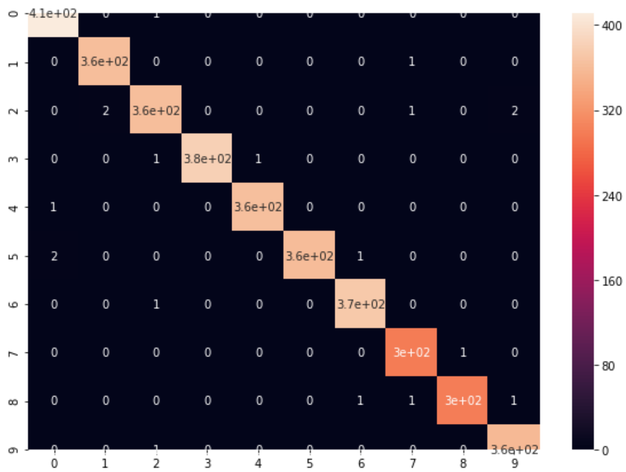
Final accuracy using logistic regression – 68%

5.3 SVM (Support Vector Machines)

SVM may not perform well when target classes are overlapping. In our case classes like talking on phone right and texting are overlapping and also reaching behind and talking to passenger. Final Accuracy for SVM is 76 %

5.4 Neural Networks (Multi Layered Perceptron)

The better performance of MLP's over ML models may be due to it ability to learn and build complex hyperplanes due to its non-linear activation units in their layers. Since, it is very likely to overfit the data we used dropout technique. Because of this nature of MLP's we have a better and higher accuracies for both training and test data sets. The confusion matrix obtained for MLP model is as below,



Final accuracy is : 98.3%

6. Contributions

6.1 Deliverables

Each team member has implemented one advanced model independently and the mapping is as follows,

Logistic Regression – Sarath Chandra Reddy (MT19037)
Support Vector Machines – Kolla Nikhil (MT19123)
Neural Networks – Mani Kumar Reddy (MT19065)

6.2 Individual Contributions

Sarath Chandra Reddy : Implementation of logistic model with histogram of oriented gradients (hog) feature extraction technique. For dimensionality reduction applied Principal component analysis.

Nikhil : Implemented SVM with various different kernels with PCA as feature reduction technique.

Mani Kumar Reddy :Implemented 2 layer neural network which included batch normalization and drop out techniques.

6.3 List of files / modules:

Sarath Chandra Reddy : basic pre-processing of the image data , hog feature extraction, implementation of logistic regression.

Nikhil : implementation of svm with pca for dimensionality reduction

Mani kumar Reddy : Building neural net model with different preprocessing and batch normalization and dropout techniques.

6.4 References

[1] B. Baheti, S. Gajre, and S. Talbar.
Detection of distracted driver using convolutional neural network.
In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1145–11456, June 2018.

[2] Y. Liang, M. L. Reyes, and J. D. Lee.
Real-time detection of driver cognitive distraction using support vector machines.
IEEE Transactions on Intelligent Transportation

[3] https://keras.io/examples/mnist_mlp/

[4] Patway, Muhammed J. A. & Parvin, Shahnaj & Akter, Subrina. (2015). Significant HOG-Histogram of Oriented Gradient Feature Selection for Human Detection. International Journal of Computer Applications. 132. 20-24. 10.5120/ijca2015907704.

[5] https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/

[6] https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60