Mrigankshi Gupta (101903005) 3COE1

Data Science Evaluation

Documentation for Distracted Driver Detection problem from preprocessing to model fitting (Notebook Link) (Github Link)

Problem Name	Distracted Driver Classification
Problem Link	https://www.kaggle.com/c/state-farm-distracted-driver-detection
Problem Type	Classification
GitHub Link	https://github.com/mrigankshi26/State-Farm-Distracted-Driver-Detection
Kaggle Link	https://www.kaggle.com/mrigugupta/distracted-driver-detection-evaluation
Libraries used	cuml, cudf,cupy,matplotlib,os,csv,time,tqdm,keras,sklearn,seaborn
Models Implemented	Logistic Regression, xgboost ,Support Vector Classifier,Lightgbm,Random Forest Classifier.
Evaluation Metrics Used	Accuracy, Confusion matrix
Kaggle Rank Achieved with total number of teams (if applicable)	NA
Proof of Rank (if applicable)	NA
Tasks done in code	- Data preprocessing - Model training and testing - Model building and evaluation - Detailed Coding documentation

Project Overview

Driving a car is a complex task, and it requires complete attention. Distracted driving is any activity that takes away the driver's attention from the road. Approximately 1.35 million people die each year as a result of road traffic crashes.

In this project our aim is to identify whether a driver is driving safely or indulged in distraction activities like texting, drinking ,calling ,etc. I have different classifiers such as **SVM ,XGBoost, Random Forest Classifier, Lightgbm,Logistic Regression** and compare their performance by tuning different hyperparameters. We evaluate the performance of these classifiers on metrics such as **Accuracy, Precision, Recall and Confusion matrix.**

Dataset

The dataset contains 22424 driver images in total downloaded from kaggle. The dataset contains coloured images of size 640 x 480 pixels which are resized to 64 X 64 coloured images for training and testing pusposes.

The 10 classes to predict are:

- Safe driving
- Texting (right hand)
- Talking on the phone (right hand)
- Texting (left hand)
- Talking on the phone (left hand)
- Operating the radio
- Drinking
- Reaching behind
- Hair and makeup
- Talking to passenger(s).

Data Preprocessing:

Here we will use pixels as features for our Training Algorithms :

1. Image is Resize to 64x64 RGB Image .
2. 64x64 coloured Image will be used for Feature Extraction Algorithms .

Algorithm Implementation : Steps

1. Dividing Train dataset into Train and Test DataSet by stratified split
2. Again we Split Train Dataset to Train and Validation Set
3. We apply Feature extraction Technique
4. Normalizing the Extracted Features using min-max Normalization
5. Dimensionality Reduction Techniques (PCA/LDA) on normalized Extracted Features
6. Combining the Extracted Features
7. We apply our Classifier and Fitting it on train_imgs and Testing it on val_imgs
8. Predicting the Class label
9. Evaluation Metrics

Observation:

There are total 22424 training samples.
The training dataset is equally balanced to a great extent and hence we do not require any downsampling of the data.

Procedure- Converting images into 64*64 .Then Encoded the object as an enumerated type or categorical variable. This method is useful for obtaining a numeric representation of an array(.factorize( )) . Then I have implemented train_test_split . I have trained the model and also tuned the model to get high accuracy.

Used Various Classifiers and got different accuracy

1. Support Vector Classifier Accuracy= 0.992417454
2. Logistic Regression Accuracy= 0.994647622
3. XGBoost Accuracy= 0.996877789
4. Lightgbm Accuracy= 0.997769833
5. Random Forest Classifier Accuracy= 0.979928612

Inference :

Lightgbm gave the best accuracy.

XGBoost performed better than Logistic Regression.

Data Augumentation and Data Extraction techniques helps the classifier to perform better.