Project Proposal
Driver Drowsiness Detection System using CNN

Team - Phoenix

Arpitha Gurumurty, Gayathri Pulagam, Surabhi Govil

Abstract

Driver Drowsiness Detection: Falling asleep at the wheel is a phenomenon which happens with 1 in every 25 adults as reported by CDC[1] . There have been reported 72000 crashes on the national highway in 2013. A number of people have been injured due to the driver's drowsiness.

This can happen not just because of lack of sleep but also the driver having sleep disorders, fatigue, consumption of alcohol, etc. Drowsiness can hamper one's ability to drive safely.

We are not just pointing out the physical state of feeling sleepy but also the brief moments where one might lose attention. To tackle this problem we aim to create a solution to detect driver drowsiness through facial feature changes and prevent accidents in the early stages.

Objective

To classify the user's attention level into one of these four categories, **Alert, Low Vigilant** or **Drowsiness** based on facial feature changes. We are planning to localize the driver's eye area which usually points to the attention level of the driver and create a Region of Interest. To do this, we plan on using Keras with CNN based architecture.

Datasets

We plan to use UTA Real-Life Drowsiness dataset[2] which contains a total of 180 videos which is about 30 hours of RGB videos of 60 healthy participants. Each participant recorded 3 different classes of video using their phone's camera or webcam : alertness (labelled as 0), low vigilance (labelled as 1) and drowsiness (labelled as 2).

The labelling emphasizes the transition from alertness to drowsiness. The participants were a combination of 51 men, 9 women from different ethnicities. In addition to this dataset, we are trying to find other datasets.

Expected results

1. By the end of the project, we expect to have a model which is able to classify a driver's attention level into 3 categories based on their facial expressions.
2. The model should be deployed to an end-point using **TFX** and should be fully functional to get inferences from the trained model.

Metrics and visualizations

1. AUC-ROC curve to evaluate the classification of the model. Because our problem is a multi-class, we are going to use **One vs All** technique[4].

2. Precision & Recall scores (using micro and macro averages)
3. Matthew's Correlation Coefficient (MCC)
4. Confusion matrix to show the combination of actual and predicted classes.
5. Cross-entropy to measure the match of actual data to the predicted probabilities.
6. Class prediction error charts to visualize the misclassified classes as a stacked bar[5].
7. To visualize the metrics and track the performance of our model we will use **TensorBoard**.

Plan for Demo (Deployment of the model)

We plan to use TensorFlow Extended (TFX) end-to-end pipeline to deploy our model and demonstrate it in real-time. We plan to deploy it as a web app for now but will work in parallel to make it iOS and Android deployable using TFLite.

Team Member Roles

1. **Arpitha Gurumurthy** - Data preparation, feature engineering, data ingestion methods, model evaluation and visualizations
2. **Gayathri Pulagam** - Feature engineering, model building, TFX model deployment, ETL for TFX, model evaluation and visualizations
3. **Surabhi Govil** - Feature engineering, model building, research, TensorBoard setup, visualizations, model evaluation

End deliverables

1. Deployed model's endpoint for testing the model in real-time.
2. Team git repository's URL with the entire code used in the project
3. Model's metrics and visualizations (TensorBoard)
4. Project Report providing the details of implementation, methodologies used, evaluations and challenges encountered during the project.
5. Presentation slide deck which will be used by the team in class to walk through the project.

References

1. https://www.cdc.gov/sleep/features/drowsy-driving.html#:~:text=An%20estimated%201%20in%2025,in%20the%20previous%2030%20days.&text=The%20National%20Highway%20Traffic%20Safety,and%20800%20deaths%20in%202013.
2. https://sites.google.com/view/utarldd/home
3. https://arxiv.org/pdf/2001.05137.pdf
4. https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/
5. https://medium.com/apprentice-journal/evaluating-multi-class-classifiers-12b2946e755b
6. https://arxiv.org/pdf/1904.07312.pdf
7. https://www.hindawi.com/journals/cin/2020/7251280/