FINAL PROJECT PROPOSAL

DRIVER DROWSINESS DETECTION

DATS 6303 - DEEP LEARNING

Professor: Dr. Amir Hossein Jafari, Ph.D.

Team Members

Aswin Balaji Thippa Ramesh, Rahul Arvind, Vishal Fulsundar

Problem Selection and Motivation

Driver drowsiness is a major cause of road accidents worldwide, often leading to serious injuries or fatalities. To help prevent such incidents, we developed a **Driver Drowsiness Detection System** that monitors a driver's eye activity (open or closed) and yawning behavior in real-time, issuing alerts when signs of fatigue are detected.

We used two Kaggle datasets for this project:

1. **Eye Gaze Detection** - to classify eye states (open/closed).
2. **Drowsiness Detection** - to detect yawning and overall drowsy behavior.

This system aims to enhance road safety by providing timely warnings, making it a practical addition to modern driver-assistance technologies.

Dataset Description

For this project, we used two publicly available image datasets from Kaggle, each designed to support key aspects of driver drowsiness detection:

1. **Drowsiness Detection Dataset**
   This dataset includes facial images categorized into four classes: open eyes, closed eyes, yawning, and not yawning. It is designed to help detect signs of fatigue by analyzing both eye state and mouth activity.

2. **Eye Gaze Detection Dataset**
   This dataset contains labeled images of eyes looking in different directions (e.g., top-left, bottom-center, middle-right). It helps in tracking eye movement and determining whether the driver is focused or distracted.

Both datasets are well-balanced and contain thousands of labeled images, making them suitable for training deep learning models for real-time drowsiness detection.

Network Architecture

Our initial approach utilizes a **Convolutional Neural Network (CNN)** trained on 2D facial inputs, specifically cropped eye and mouth regions. During training, intermediate feature maps are extracted to visualize the network's learned spatial hierarchies and confirm attention to semantically relevant areas.

For real-time inference, **OpenCV** is used to capture video frames, detect facial landmarks, and extract eye and mouth regions. Upon detecting prolonged eye closure or yawning, the system triggers an alert. To enhance performance and generalization, we plan to incorporate **pretrained architectures**, enabling transfer learning with minimal architectural adjustments.

Implementation Framework

We use **TensorFlow** with **Keras** for building and training our CNN models, benefiting from its modular design and GPU support for efficient experimentation. **OpenCV** is integrated for real-time video capture, facial landmark detection, and region extraction. TensorFlow's support for pretrained models also enables future transfer learning to boost performance.

Reference Materials and Background Sources

- **Research Papers**:
    - Driver Drowsiness Detection and Alert System https://www.researchgate.net/publication/353109539_Driver_Drowsiness_Detection_and_Alert_System

---

Performance Evaluation Metrics

Model performance is evaluated using standard classification metrics including **accuracy**, **precision**, **recall**, and **F1-score**, with particular emphasis on F1-score due to potential class imbalance. During training, we monitor validation loss and metrics to prevent overfitting.

We also visualize confusion matrices and ROC curves to better understand classification behavior and identify areas for improvement. Future iterations may include temporal metrics or video-based sequence modeling to further enhance evaluation.

Project Timeline

Week 1: Perform data, image preprocessing, and train the initial CNN model for eye and mouth movement classification.
Week 2: Integrate OpenCV for real-time video capture, facial landmark detection, and model inference.
Week 3: Optimize the model, evaluate performance, and complete deployment with real-time alert functionality.