**Project – Driver Drowsiness Detection System**

<https://github.com/Badhrinadh0987/ECE510_HW_AI-ML>

https://github.com/Badhrinadh0987/ECE510\_HW\_AI-ML/tree/main/My\_Progress  
  
  
**Heilmeier's Catechism: Driver Drowsiness Detection System**

**1. What are you trying to do?**  
Build a real-time system that detects driver drowsiness using webcam input by analyzing facial cues such as eye closure and yawning to help prevent accidents due to fatigue.

**2. How is it done today, and what are the limits of current practice?**  
Most modern systems use expensive hardware like infrared sensors, steering behavior analysis, or are embedded in premium vehicles. These solutions are not accessible or affordable for regular users. Current software-based solutions may not work well in real-time or under diverse lighting conditions.

**3. What’s new in your approach and why do you think it will be successful?**  
We're using a **custom-built CNN model trained on webcam images** for lightweight real-time performance. The use of **facial landmarks and deep learning** allows the system to operate using just a regular webcam, making it cost-effective and accessible for broader implementation.

**4. Who cares?**  
This is critical for drivers, especially those in transportation industries (e.g., truck drivers, cab services), as well as everyday commuters. Car manufacturers, fleet operators, and public safety organizations would also benefit from affordable drowsiness detection solutions.

**5. If you are successful, what difference will it make?**  
It can reduce fatigue-related accidents by providing real-time warnings, potentially saving lives and improving road safety. It could also lead to wider adoption of drowsiness detection in low-cost vehicles or even as a mobile app.

**6. What are the risks and the payoffs?**  
**Risks**:

* + Variability in lighting or face angles may affect accuracy
  + Custom dataset may be too small for generalization
  + Real-time processing needs to be optimized for performance

**Payoffs**:

* + Affordable, software-based safety tool for all drivers
  + Scalable and customizable for multiple platforms (PC, mobile, car systems)

**7. How much will it cost?**  
Very minimal — just a computer with a webcam and open-source libraries (OpenCV, TensorFlow). No special hardware is required.

**8. How long will it take?**

* Data collection: 1–2 days
* CNN model development & training: 1–2 weeks
* Real-time integration and testing: 1–2 weeks  
  Total: ~3–4 weeks for a working prototype

**9. What are the midterm and final “exams” to check for success?**

* **Midterm**: Successfully classify "alert" vs. "drowsy" on test data with acceptable accuracy (>85%)
* **Final**: Real-time detection from webcam with consistent alerts and minimal false positives/negatives

04/16/2025

* Figure out the boundaries of SW/HW.   
  Software -> SW with HW accelerator -> Communication cost -> Hardware execution time
* Workflow/toolchain

1. PyUVM
2. Cadence
3. Synopsys

Objective: Codefest2

CNT, Neuromorphic architectures  
  
pick - AI/ML algorithm/workload

Non-conventional workload  
Google scholar – to find relevant literature

Accelerator chiplet for that workload  
  
Design Trade-offs?  
 Performance

Flexibility (general purpose vs app specific)

Energy efficiency

Power Density (avoid burning chips)

Complexity

Why design new HW if there is already one available to do the job? E.g., a RISC processor?

* From a flexibility and design effort perspective, as much as possible should be in software

**AI vs ML:**

* All ML is AI, but not all AI is ML
* AI is the destination (e.g., intelligent machines), while ML is one path to get there (learning from data)

Abstract neuron:

* ReLU – No upper limit, zero parameter
* When, How and why to use a neural network
* Inference mode after training the network without modifying the weights
* How do we unlur the neural network
* Train the network using backward propagation which automatically calculates the error and updates the network
* Supervised training – Tells you got the error
* Semi supervised training
* Unsupervised training

General AI accelerator architecture