Technology Bucket: **Smart Vehicles**
Company Name/ Ministry Name: **ARAI**
Team Leader Name: **Adwait Bhope**

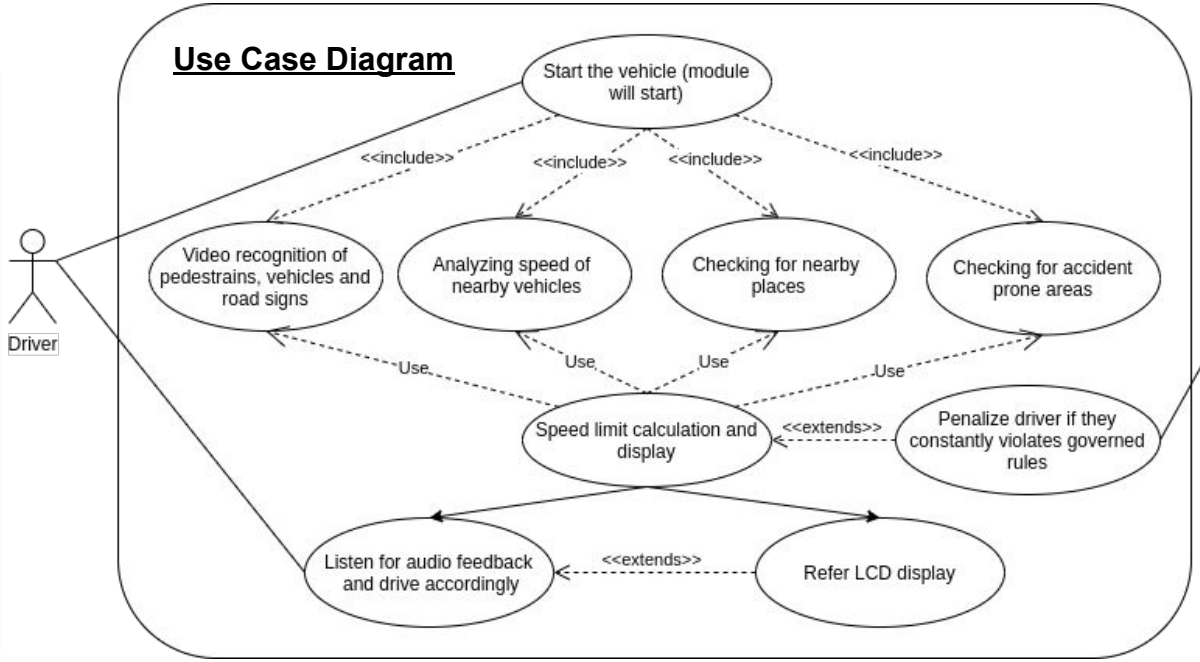
Category: **Software**
Problem Code: **UK149**
AICTE College Code: **1-3508330757**

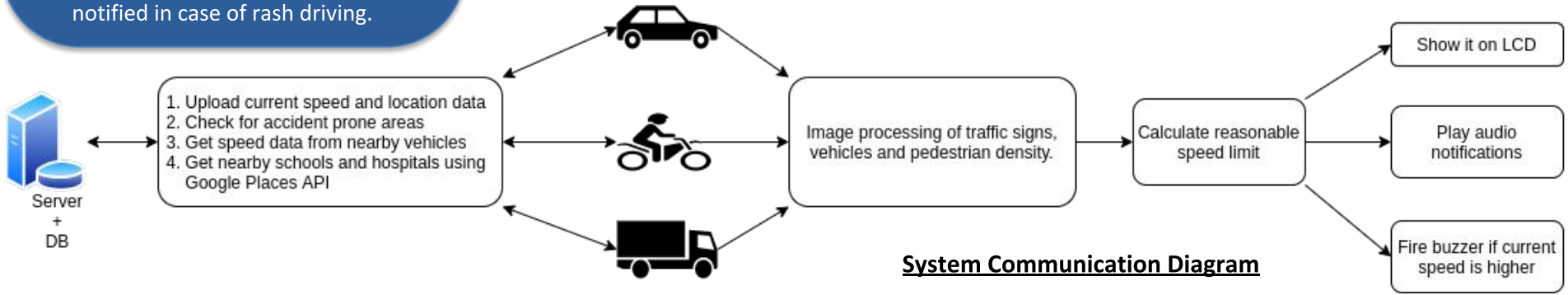
Idea / Approach Details

- The solution comprises of a device with a **camera** mounted in front of the vehicle, connected to a microprocessor like **Raspberry Pi / Jetson Nano**. An **LCD** screen is used along with an **audio assistant** for alerts and notifications. It will include a **GSM/GPRS/GPS module** to get the **live location of the vehicle** and calculate **current speed**, and an **accelerometer + gyro sensors** to detect jerks. The system will be powered by solar panels (a backup can be provided by the vehicle's battery).
- **Pedestrian safety** will be ensured by calculating a **reasonable speed limit** by using the following factors:
    1. **Video Recognition** based on live feed from the camera: Detecting **road signs** (according to IRC standards)**, pedestrians and vehicle density** in **real-time** (using Convolutional Neural Networks). This will be performed **locally, offline**.
    2. **Nearby Places**: **Schools** and **hospitals** near the user's **current location** will be checked using **Google Places API** and **road signs**.
    3. **Analyzing neighbouring vehicles**: Other vehicles (using our system) will **report data to the server**. This will be used to **track the speed** of nearby vehicles travelling in the same direction at that instant.
    4. **Accident-prone areas**: Nearby accident-prone areas will be checked **based on current location** and the user will be **alerted**
- These **factors are independent** of each other. Hence, others will contribute even if one of them is unavailable. For example, even if it is a place with no internet connectivity, video recognition will still work as it is **performed offline**.
- The recommended speed limit will be calculated dynamically and shown on the **LCD**
    - The display will be **color-coded** for easy interpretation (**red** if exceeding the speed limit, otherwise **green**)
    - **Audio notification** will be given to avoid distraction
    - A **buzzer** will be used in case of alerts (like crossing speed limit, accident-prone areas)
- The system will **start automatically** when the car starts. Video Recognition will be performed by using **Transfer Learning** on **Inception SSD V2** pre-trained model. The size of the generated bounding boxes will be used to **calculate the distance** of the objects. **Expanding or shrinking boxes** will be used to detect if objects are **approaching or receding** from the car, respectively.
- Data from **road signs** will be extracted using **Image Processing** to track governed speed limit and use that as an absolute upper bound.
- The system will **keep communicating** its speed, location **with the server**. The server will **analyze data**, **check for accident-prone areas** and **nearby schools and hospitals**. It will send this **information back to the vehicle** system immediately.
- A formula will be used to calculate a final speed limit considering all this data.

- **Accelerometer + gyro sensors** will be used to detect when an **accident** takes place. It will be reported to the server. A **database of accident prone areas** will be built gradually. This will be used to identify if vehicles are travelling through such an area and the driver will be alerted.
- If the driver violates the governed speed limit often, he can be penalized accordingly (by generating a challan / automatic deduction from bank account).
- For commercial vehicles (buses like Volvo), concerned authorities can be notified in case of rash driving.

Use Case Diagram





System Communication Diagram

Technology Stack

Client
Google Places API
OpenCV, TensorFlow (for video recognition)
Raspberry Pi + Camera + GSM/GPRS/GPS module (GPS to get live location and GSM/GPRS for internet)

Server
gunicorn + nginx
Flask (Python)
MongoDB (NoSQL database)

Advantages

- Users **don't have to interact** with the system at all
- **Edge Computing** is used for Video Recognition, so it will work offline. It does not require internet.
- Even if places are not located on Google Maps, they can be detected using road signs.
- **Dedicated hardware is better than mobile app** as users may not use the app. Mobile phones will need to be mounted all the time. Further, some phones may not provide optimal performance (e.g. bad camera, slow processing, low battery).
- Can also be used with **2 wheelers, heavy vehicles, etc.**
- The entire system can be targeted for **sale to vehicle manufacturers** to fit inside their cars; this will result in **widespread adoption** of the system. It can also be **retrofitted to existing vehicles**.
- **System performs better** when the number of vehicles with this module grow, as **more data** will be available

Showstoppers

- Internet/GPS connectivity issues in cloudy environment (Video Recognition will work)
- Low visibility (for eg. fog) can affect real time recognition (IR camera / ultrasonic sensors can be used)
- Hardware constraints (Powerful boards like NVIDIA's Jetson Nano can be used)