DRIZY- Collaborative Driver Assistance Over Wireless Networks

Nakul Garg[1], Ishani Janveja[1], Divyansh Malhotra[1], Chetan Chawla[1], Pulkit Gupta[1], Harshil Bansal[1], Aakanksha Chowdhery[2], Prerana Mukherjee[3], Brejesh Lall[3]

[1]Bharati Vidyapeeth's College of Engineering Delhi [2]Princeton University [3]IIT Delhi

Objective

- Road accidents are a major cause of human fatalities (1.2 million fatalities each year).
- We design a driver assistance system that uses location data and dashboard camera feed. Edge computing on smartphone and local processing on Raspberry Pi combine to alert drivers and nearby vehicles of potential collisions with pedestrians and other vehicles at intersections.
- Each vehicle uploads its location sensor data from smartphones when it is near an intersection (over wireless networks). Cloud platform uses the vehicular locations to predict potential collisions at intersections and send alerts to drivers ahead of time.
- Optimized local processing on dashboard camera video feed detects pedestrians in collision trajectory in real time.
- We benchmark wireless network latencies and traffic densities when such collaborative driver assistance is feasible using edge/cloud computing framework.

System Design

Sensing on vehicle. On-board camera detects pedestrians at 9.8 fps using Raspberry Pi. DRIZY mobile application infers speed and location attributes of the vehicle.

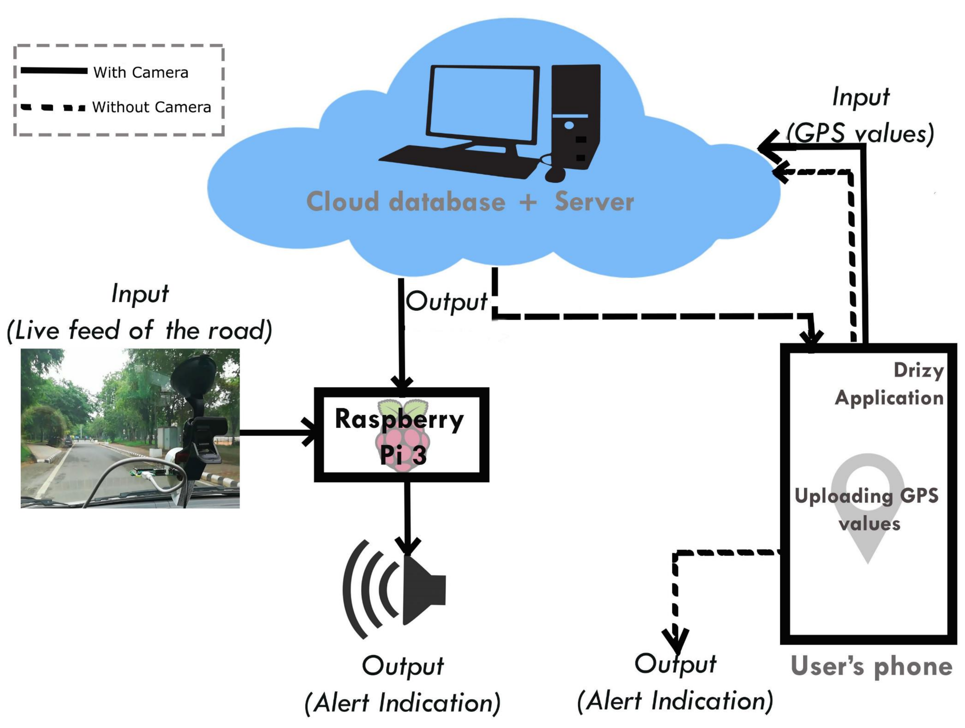

Figure 1: DRIZY Framework with GPS sensors and a dashboard camera (optional).

Accident blackspots. Accident prone areas whose GPS coordinates are known to the server. **Cloud server.** Cars are dynamically assigned to clusters in the vicinity of each accident blackspot. Each cluster uploads vehicular data to the cloud database. The server then predicts potential vehicle-to-vehicle collisions at road intersections and sends alerts to concerned vehicles.

Alerts. Smartphone application or Raspberry Pi generates different sound alerts for potential collisions to pedestrians and other vehicles.

Vehicle-to-vehicle collisions at road intersections

Each vehicle that enters an accident blackspot uploads its GPS coordinates via a smartphone application to the cloud database. The database maintains an optimized hierarchical structure of accident blackspots, associated roads, the vehicles and their attributes.

- Speed of car, Latitude, Longitude
- Direction of vehicle
- Road Number

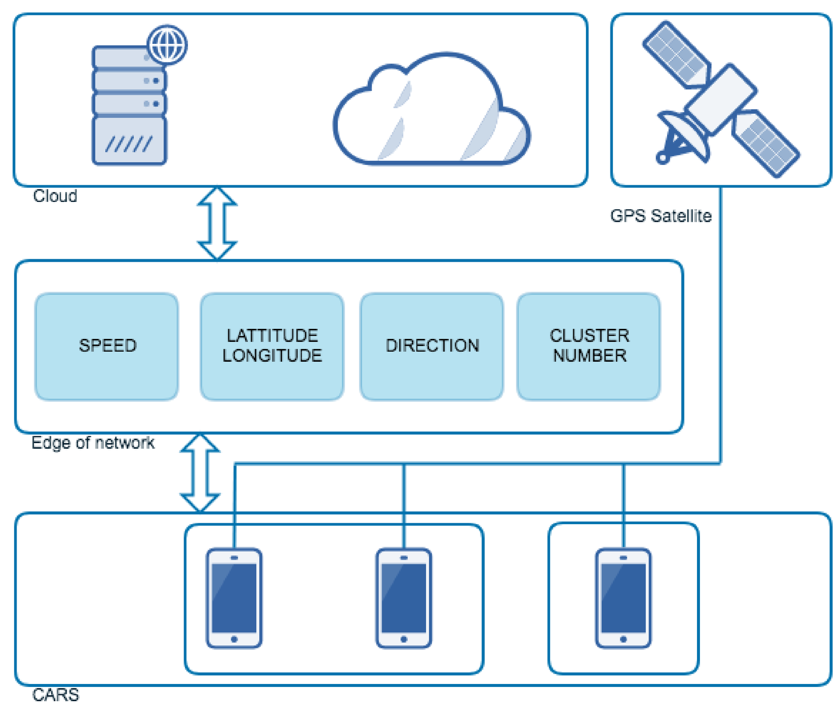

Figure 2: Vehicle-to-vehicle collision avoidance at road intersections: Edge-computing on smartphone only uploads GPS coordinates to the relevant accident blackspot in cloud database.

Cloud server uses these attributes in database to predict potential collisions in a blackspot every few seconds and alerts the vehicles on a probable collision trajectory.

Vehicle-to-pedestrian collisions

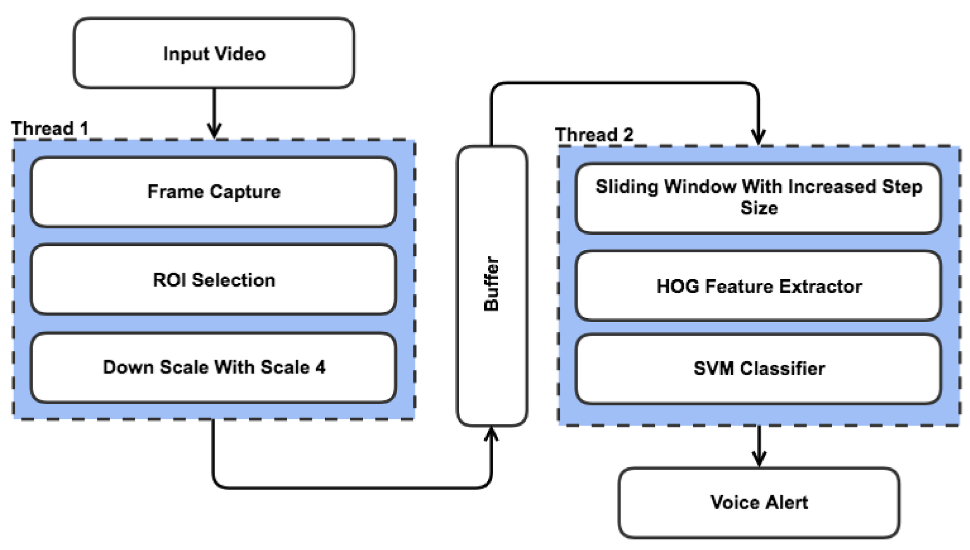

Figure 3: Multi-thread processing to detect pedestrians on embedded platform.

A monocular dashboard camera captures the view in front of the vehicle. To design a cost-effective edge solution, we process the video feed on Raspberry Pi and optimize the processing to detect pedestrians in real-time. The camera video feed is sent to Raspberry Pi GPU via the CSI port. We then detect pedestrians in a region of interest using a HOG/SVM classifier. We use multi-thread processing in parallel to process video frames to a camera buffer and pedestrian detection on multiple frames. Thus, pedestrians can be detected in real-time at 9.8 fps despite limited compute capability of Raspberry Pi.

Evaluation

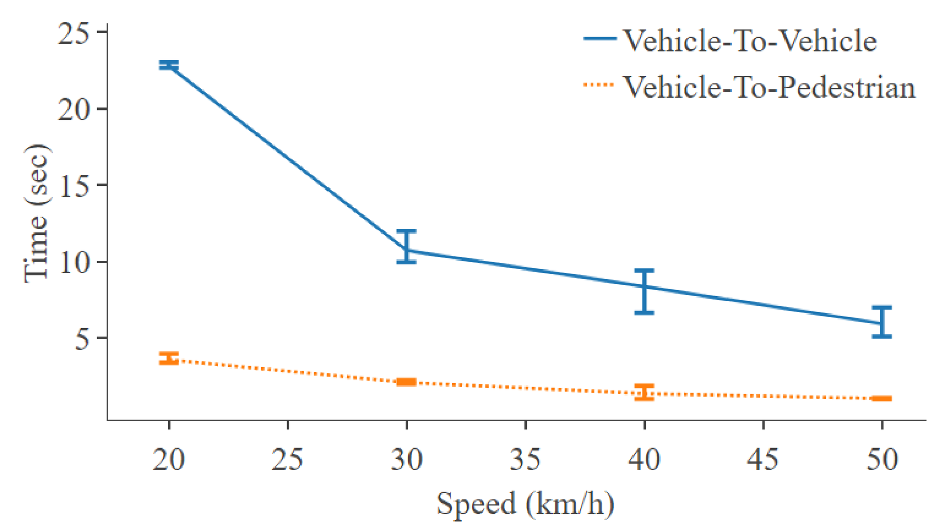

Figure 4: Available reaction time before collision with pedestrians and vehicles. Error bars show maximum and minimum reaction times during 50 test drives with vehicles approaching an intersection point or a pedestrian.

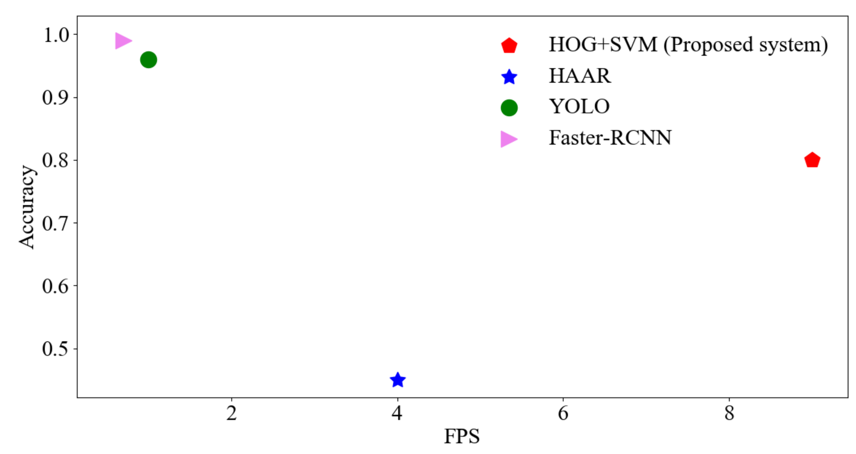

Figure 5: Precision vs frames processed per second for different algorithms in pedestrian detection module. Video dataset collected on Indian roads with 5500 positive images (pedestrian) and 4500 negative (non-pedestrian) images.

Summary and conclusions

For vehicle-to-vehicle collisions, edge computing on the smartphone application processes location sensor data to identify accident blackspot cluster so that the cloud database only predicts collisions within each blackspot, reducing processing latency at the server. For vehicle-to-pedestrian collisions, edge computing on the embedded platform optimizes processing of the video feeds to enable pedestrian detection in real-time. Under wireless network latencies, evaluation suggests that **DRIZY** is feasible in low and medium traffic density conditions.