Poster: DRIZY- Collaborative Driver Assistance Over Wireless Networks

Nakul Garg*[1], Ishani Janveja*, Divyansh Malhotra*, Chetan Chawla*, Pulkit Gupta*,

Harshil Bansal*, Aakanksha Chowdhery†, Prerana Mukherjee††, Brejesh Lall††

* Bharati Vidyapeeth's College of Engineering Delhi, †Princeton University, †† IIT Delhi
[1]nakulgarg.2208@gmail.com, † aakanksha@princeton.edu, †† {eez138300, brejesh}@ee.iitd.ac.in

ABSTRACT

Driver assistance systems, that rely on vehicular sensors such as cameras, LIDAR and other on-board diagnostic sensors, have progressed rapidly in recent years to increase road safety. Road conditions in developing countries like India are chaotic where roads are not well maintained and thus vehicular sensors alone do not suffice in detecting impending collisions. In this paper, we investigate a collaborative driver assistance system "DRIZY:DRIve eaSY" for such scenarios where inference is drawn from on-board camera feed to alert drivers of obstacles ahead and the cloud uses GPS sensor data uploaded by all vehicles to alert drivers of vehicles in potential collision trajectory. Thus, we combine computer vision and vehicle-to-cloud communication to create comprehensive situational awareness. We prototype our system to consider two types of collisions: vehicle-to-vehicle collisions based on uploading GPS sensor data of vehicles to cloud and vehicle-to-pedestrian collisions based on detecting pedestrians from vehicle's dashboard camera feed. Sensor data processing in each vehicle occurs on smartphone for GPS values which are then uploaded to cloud and on raspberry pi3 for video feeds to make a cost-effective solution. Experiments over both 4G and wireless networks in India show that collaborative driver assistance is feasible in low traffic density within acceptable driver reaction time of <5 sec, but can be limited by the time to process compute-intensive video feeds in real-time. We investigate novel ways to optimize the processing to find an acceptable trade-off.

CCS CONCEPTS

• **Networks** → **Location based services**; • **Computing methodologies** → **Image processing**; • **Computer systems organization** → *Sensor networks*; *Real-time system architecture*;

KEYWORDS

Driver assistance system; vehicle-to-cloud communication; edge computing; pedestrian detection; HOG+SVM sliding-window optimization

MobiCom '17, October 16–20, 2017, Snowbird, UT, USA

ACM ISBN 978-1-4503-4916-1/17/10.
https://doi.org/10.1145/3117811.3131255

1 INTRODUCTION

Road accidents continue to be a major cause of human fatalities, resulting in as many as 1.2 million fatalities each year [2]. In developing countries, such as India, two major causes of road accidents are vehicle-to-vehicle collisions at road intersections and vehicle-to-pedestrian collisions when drivers are distracted. Such accident prone areas are referred to as "accident blackspots".

Advanced driver assistance systems (ADAS) intend to increase the road safety by assisting in crucial decision-making using vehicular sensors, such as CMOS vision sensors, LIDARs, etc. Road conditions in developing countries, such as India, include chaotic traffic on unmaintained roads. Pedestrian behavior is particularly unpredictable and people cross roads anywhere without crosswalk. Such conditions make the design of driver assistance systems particularly challenging and on-board vehicular sensors alone do not suffice to predict impending collisions.

In this paper, we propose a collaborative driver assistance system DRIZY in which vehicular sensor data and inference drawn from the data is uploaded over a wireless network to the cloud which alerts the vehicles of obstacles in potential collision trajectory. The objective of our system is twofold: i) to detect the presence of pedestrians in the specific region of interest around the host vehicle using dashboard camera feed and alert the driver, and ii) to predict and alert vehicles on a collision course using GPS sensor data uploaded by all vehicles. We delegate pedestrian detection processing to an embedded platform and a dedicated application in the smartphone in each vehicle handles the impending collision alerts. The key challenges in our system design are ensuring low latency in estimating and sending alerts about vehicle-to-vehicle collisions and ensuring real-time processing to detect pedestrians. Our system design solves these challenges by pre-processing the sensor data locally on smartphone/embedded platform and minimizing overhead processing on the cloud. Experiments over 4G and WiFi networks in India show that collaborative driver assistance is feasible in low traffic density within acceptable driver reaction time of <5 sec.Further, we optimize the pedestrian detection module on embedded platforms to achieve real-time processing at 9.8 fps.

2 SYSTEM DESIGN

We discuss key components of the system architecture below and illustrate the same in Fig. 1.

Sensing on the vehicle. The driver installs DRIZY app which processes GPS values obtained from smartphone to infer the vehicle location, speed and direction of travel. In addition, the vehicle can be equipped with a dashboard camera that captures video-feed of the view in front of the car and relays it to an embedded platform to provide real-time assistance warnings when a pedestrian is in collision trajectory.

Cloud database and server. DRIZY app uploads processed vehicular data to the cloud database. The cloud server processes data for each accident prone area to predict vehicles in collision trajectory and ping alerts to the app in case of impending danger.

Local processing vs cloud processing. DRIZY app pre-processes the vehicular GPS data to identify if the vehicle is close to an accident prone area and to add it to a "cluster" of vehicles present in the same area, also called accident blackspot. Thus, the cloud only needs to process data captured in each accident blackspot in real-time to predict potential collisions.

Alarm Unit. The embedded platform and cloud server generate alerts for vehicle and pedestrian collisions respectively. In a with-camera system, both alerts are sounded from a piezo-buzzer attached to the embedded system. Whereas the vehicle collision alerts are received on the smartphone itself for users using app only. We next discuss how we infer the two type of collisions: a) vehicle-to-vehicle collisions at road intersections, and b) vehicle-to-pedestrian collision on roads.

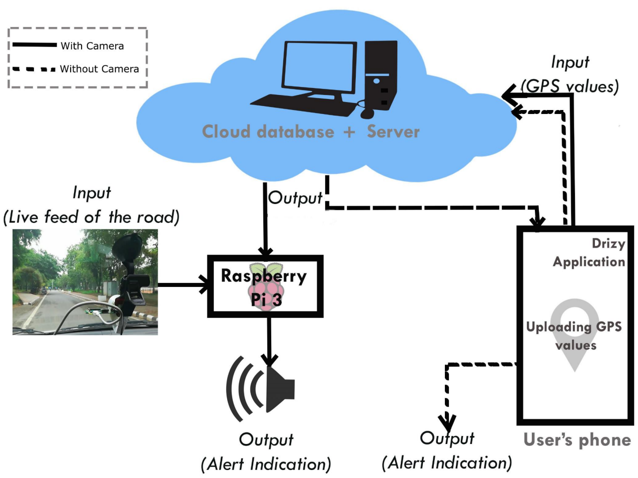

Figure 1: Workflow of the DRIZY Framework with and without camera.

2.1 Vehicle-to-vehicle collisions at road intersections

The cloud maintains a database representation of all accident blackspots in the form of hierarchical representation. DRIZY app uploads processed GPS data collected by smartphone to the respective blackspot in database. This data includes GPS coordinates, road number, speed and direction of motion towards or away from intersection point. For each accident blackspot, the cloud then predicts the collisions of vehicles approaching the same road intersection. Algorithm 1 discusses how the server predicts impending collisions and calculates time for each vehicle to reach the intersection point. If the difference in estimated time of any two vehicles to approach the intersection is less than a threshold set to 5 secs, then the cloud sends an alert to vehicles of interest. The algorithm adapts its threshold according to the speed of the vehicle. For example, if the vehicle is travelling too fast, then the threshold is doubled to 10 secs and prior alerts are generated, thereby giving the driver sufficient time to apply brakes after alert reception.

Algorithm 1: Server Code Algorithm

```
foreach pair of vehicles in Accident Black Spot do
    Get vehicle's GPS, speed, road number from Cloud
    if roadofvehicle1 != roadofvehicle2 then
        if direction of both vehicles is toward intersection then
            Find the distance of two vehicles from intersection
              using Haversine formula [5]
            Predict times t1 and t2 to reach the intersection
              using linear predictive model
            if |t1 − t2| < threshold then
                Send collision alert to vehicles
            end
        end
    end
end
```

2.2 Vehicle-to-pedestrian collisions

The embedded platform uses computer vision to detect pedestrians in the video feed collected from dashboard camera. This is done in real-time on using sliding window algorithm with HOG+SVM classifier. This feature is aimed specifically to help distracted drivers stay aware. Computationally intensive classifiers in computer vision algorithms limit the video processing speed on embedded platforms, such as Raspberry Pi 3. We optimize this processing using several steps. We select region of interest based on known geometric constraints and reduce the number of sliding windows where we detect pedestrians. Thus, the processing time for each frame is reduced by a factor of 7x (from 1.4 fps to 9.8 fps). Next, we use multithreading to parallelize the capture of camera frames and pedestrian detection for multiple frames. Finally, we speed up the input of video feed by using Raspberry Pi camera module that connects directly to GPU instead of relying on USB camera. Thus, novelty of this feature lies in the optimisations performed to minimize computation and increase available reaction time for drivers.

3 EVALUATION

3.1 Vehicle-to-vehicle collisions at road intersections

This section evaluates 2 parameters: (i) available time for driver to apply brake before collision (ii) frequency of alerts generated by DRIZY. We compare DRIZY with an "Always On" system that gives alerts to every vehicle in the accident blackspot.

Experiment. We perform an experiment at IIT Delhi campus roads to analyze the reaction time with 50 test drives and 2 vehicles approaching an intersection point. With DRIZY installed in them we find the available reaction time after reception of alerts. Figure 2 shows a plot between speed and reaction time available. For speeds limits between 20-25 kmph the system generates alerts with available reaction time >20s and for speed limits reaching 40-45

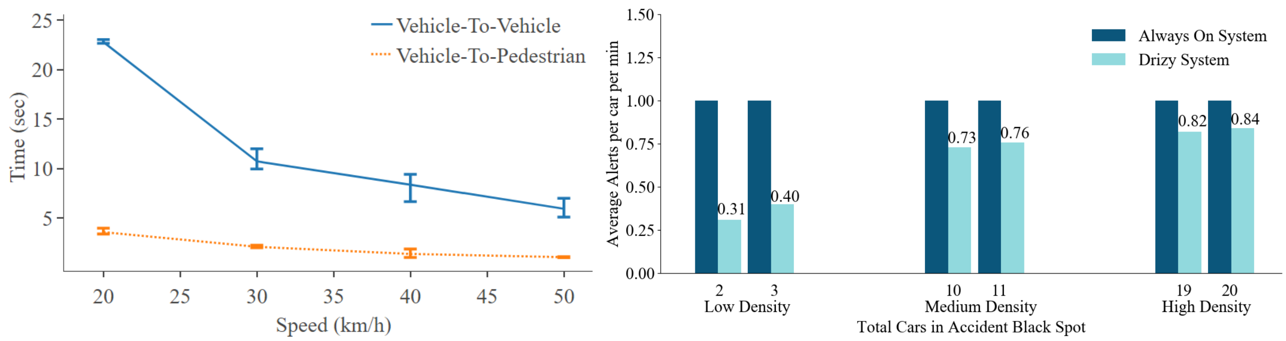

Figure 2: Left: Available Reaction time before collision with pedestrians and vehicles. It shows maximum and minimum values of time obtained during test-runs. Right: Comparative analysis of a system that gives alerts to every vehicle in the accident prone area with DRIZY that predicts collisions before giving alerts. The values were obtained using simulation and spawning of vehicles with random attributes in an accident blackspot.

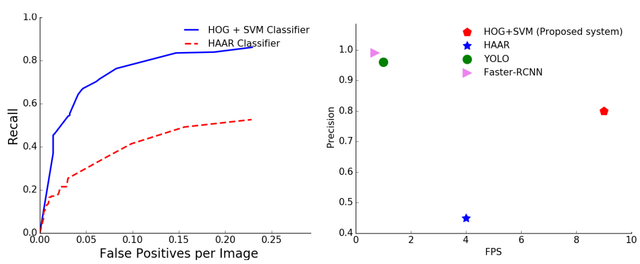

Figure 3: Left: Recall vs FPPI for pedestrian classifiers at varying thresholds. Right: Precision vs frames processed per second for various detection techniques.

kmph, it is found to be close to 8s. Latencies associated with upload, download and server processing over 4G and WiFi networks are shown in in Table 1. Due to the small packet size of the data being uploaded, the throughput of the system does not dependent on the network bandwidth and just requires connectivity.

Simulation. In order to study the frequency of collision alerts in different traffic densities, we simulate an experiment. We generate synthetic data with uniform distribution for 'N'vehicles which includes vehicle distance from intersection ranging between 0-400m, road number from 0-4, vehicle speed in range 1-36kmph and direction of motion of vehicle. This data is then accessed by the server to predict collisions. Figure 2 shows the comparison of DRIZY with an Always On system at low, medium and high density traffic levels. The proposed system reduces the redundant warnings by 80%, 50% and 10% in low, medium and high traffic density, respectively.

	4G Time	Wifi Time
Upload/Download	258/746 ms	229/678 ms
Server processing	30 ms	30 ms

Table 1: Average latencies in Collaborative Assistance workflow over different network types

3.2 Vehicle-to-pedestrian collisions on roads

This section addresses the question of how well Drizy can detect pedestrians keeping sufficient reaction time for the driver.

Experiment. We analyze the accuracy of proposed technique against various state of the art pedestrian detection techniques on video dataset collected on Indian roads with 5500 positive images (pedestrian) and 4500 negative (non-pedestrian/background) images. Similar set of test drives as in the case vehicle-to-vehicle collision are repeated for determining the trend of available reaction time after vehicle-to-pedestrian collision alerts.

Result. The plots of obtained results are shown in Fig. 2, 3. HOG[1]+ SVM classifier (proposed system) provides a speedup 10x over faster RCNN[4] and YOLO[3] at high precision rates. It is observed that HOG+SVM classifier performs better over HAAR[6] with a gain of 0.4% recall at 0.1 FPPI (false positives per image).The pedestrian detection module is capable of successful detection upto a range of 18 to 23m in varied lighting conditions during the day. The system warns the driver 4-5 seconds prior to approaching a pedestrian ahead at 10-40 kmph speed limit.

4 CONCLUSIONS

Results validate that a collaborative driver assistance system is feasible in low and medium traffic density. While video feed based assistance turns out to be useful in clear weather conditions the alerts inferred on processing GPS sensor data over cloud find their utility even in bad weather conditions. They are also capable of avoiding collisions at unsupervised intersections especially in low traffic areas where drivers tend to drive carelessly. In our future work we aim to implement forward collision alert to avoid tailgating and embed the concept of a "Green Corridor" to assist an approaching ambulance. We would also consider peer-to-peer interactions to further reduce the latency in generating assistance warnings.

REFERENCES

[1] Navneet Dalal and Bill Triggs. [n. d.]. Histograms of oriented gradients for human detection. In *CVPR 2005*.

[2] World Health Organization. 2015. *Global status report on road safety 2015*. World Health Organization.

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. [n. d.]. You only look once: Unified, real-time object detection. In *CVPR 2016*.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. [n. d.]. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS 2015*.

[5] C Carl Robusto. 1957. The cosine-haversine formula. *The American Mathematical Monthly* 64, 1 (1957), 38–40.

[6] Paul Viola and Michael Jones. [n. d.]. Rapid object detection using a boosted cascade of simple features. In *CVPR 2001*.