Poster Abstract:
SmartRoad: A Crowd-Sourced Traffic Regulator Detection and Identification System

Shaohan Hu, Lu Su, Hengchang Liu, Hongyan Wang, and Tarek F. Abdelzaher
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, 61801, USA
{shu17, lusu2, hl4d, wang44, zaher}@illinois.edu

ABSTRACT
In this paper we present SmartRoad, a crowd-sourced sensing system that detects and identifies traffic regulators, traffic lights and stop signs in particular. As an alternative to expensive road surveys, SmartRoad works on participatory sensing data collected from GPS sensors from in-vehicle smartphones. The resulting traffic regulator information can be used for many assisted-driving or navigation systems. We implement SmartRoad on a vehicular smartphone testbed, and deploy on 35 external volunteer users' vehicles for two months. Experiment results show that SmartRoad can robustly, effectively and efficiently carry out its detection and identification tasks without consuming excessive communication energy/bandwidth or requiring too much ground truth information.

Categories and Subject Descriptors
C.5.3 [**Computer System Implementation**]: Microcomputers—*Portable devices*

General Terms
Algorithms, Design, Performance

Keywords
Crowd Source, Road Sensing, Traffic Regulator

1. INTRODUCTION
Traffic regulators, such as stop signs and traffic lights, are designed to regulate competing flows of traffic at intersections. They are among the most commonly used traffic control signals, and play significant roles in people's daily driving behaviors. Despite the safety and convenience benefits they bring, the stop signs and traffic lights do charge their toll. The stop-and-go movement pattern of vehicles caused have resulted in substantial increase of gas consumption and CO_2 emissions. Driven by this problem, some recent efforts are taken to reduce the negative effects, such as GreenGPS [3]. Navigation services like this need to take into account the actual locations of traffic lights and stop signs. Yet, unlike the case with road-maps, no nationwide database exists today that documents traffic light and stop sign locations. Instead, this information is quite fragmented, buried in physical archives of different counties and municipalities.

To address the above challenge, in this paper, we develop a novel crowd-sourced traffic regulator detection and identification system, called SmartRoad, that can automatically detect and identify stop signs and traffic lights from participatory sensing data shared by individuals from their vehicles.

The general design of SmartRoad follows a client-server framework. We place the client component on a vehicular smartphone testbed, and the server component on a workstation. Compared with dedicated sensing, computing, or communication devices, smartphones are more suitable for large deployment due to their popularity. Users can easily download and install SmartRoad just like any other normal mobile applications. We implement SmartRoad on a vehicular smartphone testbed, and deploy it on 35 external volunteer users' vehicles. Through an experiment of two months collecting around 4000 miles of driving data containing hundreds of regulator-controlled and uncontrolled locations, we demonstrate that SmartRoad can deliver outstanding detection and identification performance without consuming excessive communication energy/bandwidth or requiring too much ground truth information.

2. SYSTEM ARCHITECTURE
In this section, we provide an overview of our SmartRoad participatory sensing system that carries out the traffic regulator detection and identification tasks. SmartRoad contains three modules: a data acquisition module, a detection and identification module, and a feedback module. They are deployed on two different platforms: distributed in-vehicle deployed smartphones, and a central server. Figure 1 illustrates the architecture overview of the SmartRoad system. We next discuss each of these three modules in more detail.

2.1 Data Acquisition Module
The data acquisition module is implemented on Google's Galaxy Nexus Android phones, equipped with 1.2GHz dual core CPU, 1GB memory, and 16GB flash storage, running on Android 4.0 operating system. We collect readings from the following phone sensors: i) *GPS Sensor*, the main source of the data to be used for our detection and identification tasks. Every single GPS reading includes the instantaneous latitude-longitude location, speed, and bearing of the vehi-

IPSN'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.
ACM 978-1-4503-1959-1/13/04.

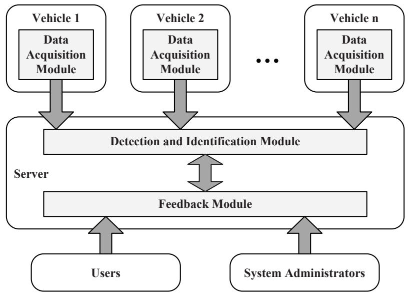


Figure 1: Architecture of the SmartRoad system

cle; and ii) *Power Sensor*, which reflects the car's engine on/off status, is used to start or stop data collection and communication.

Given the GPS data, we extract 5 features that can characterize the different driving patterns generally displayed and observed for the differently regulator-typed intersections:

Final Stop Duration: This feature captures the time duration of the last stop that a car makes in front of an intersection before crossing.

Minimum Crossing Speed: This feature represents the lowest speed at which a car crosses an intersection.

Number of Decelerations: This feature captures the number of times that a car decelerates as it approaches an intersection.

Number of Stops: This feature captures the number of stopping actions in the road segment between two intersections.

Distance from Intersection: This feature measures the distance between the intersection coordinate location and the point where a car makes its last stop, if any, before crossing the intersection.

2.2 Detection and Identification Module

The detection and identification module resides on the central server. It takes as input the extracted features together with a training set of intersections whose ground truth traffic regulator information are known, and outputs the label indicating the type of traffic regulator for each intersection. We use random forest [1] as our base classifier. Random forest is a decision tree based classification algorithm that trains multiple decision trees simultaneously and has them vote for the final classification decision. In practice, ground truth label information is limited and expensive to acquire, thus we consider a realistic scenario where initially only a tiny amount of training data is available, with further label information being acquired incrementally either on demand or opportunistically. To leverage this, we design and implement two adaptive mechanisms as follows:

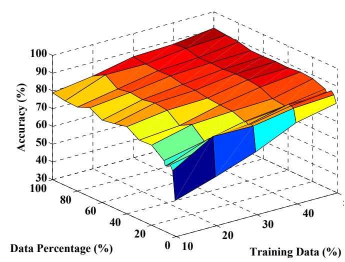
Active Learning Adapter: In realistic participatory sensing applications, it is feasible that budget allows to manually acquire ground truth information, but only up to some small amount compared to the size of the entire sensing task. Thus, an important question to ask here would be, for which intersections should we pay to get their ground truth information in order to maximize our final system detection and identification performance? To answer this question, we propose an *Active Learning Adapter*, which looks at the past classification results, identify the intersections for which the classification algorithms are the least confident about, and then hire people to manually acquire the ground truth information for these particular intersections. Here we borrow the active learning philosophy [2].

Self Training Adapter: The *Self Training Adapter*, which adopts the idea of self training [4], looks at its own past classification results and try to take advantage of them to improve system performance. More specifically, the classified intersections that have the highest confidence scores from the classifiers are progressively collected and added to the training set. The intuition behind is that classification results with high classification confidences are most likely to be correct, and thus including these data points into the training set will likely help expedite the overall classification tasks.

2.3 Feedback Module

The feedback module also resides on the central server. It visually presents detection and identification results via a web service interface. The web service interface can also be used by system administrators and users to correct detection errors or provide ground truth information, which is then sent back to the detection and identification module for dynamic and adaptive performance improvement.

3. PERFORMANCE EVALUATION



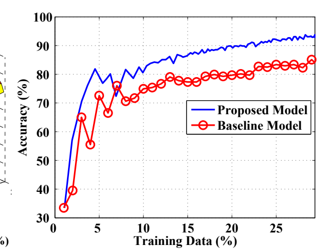


Figure 2: Classification with Active Learning and Self Training Deactivated

Figure 3: Classification with Active Learning and Self Training Activated

Figure 2 shows the classification performance when the active learning and self training adapters are deactivated. As can be seen, as the amount of data, or labeled training data, or both, increases, classification performance also improves.

The classification results with two adapters activated are shown in Figure 3. For comparison, we design a baseline classification model trained from the same amount of training data, which are selected at random. As one can see, the proposed method outperforms the baseline scheme at all the iterations.

4. REFERENCES

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] S. Dasgupta and J. Langford. A tutorial on active learning.

[3] R. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. Abdelzaher. GreenGPS: A participatory sensing fuel-efficient maps application. In *MobiSys*, 2010.

[4] X. Zhu. Semi-supervised learning literature survey, 2006.