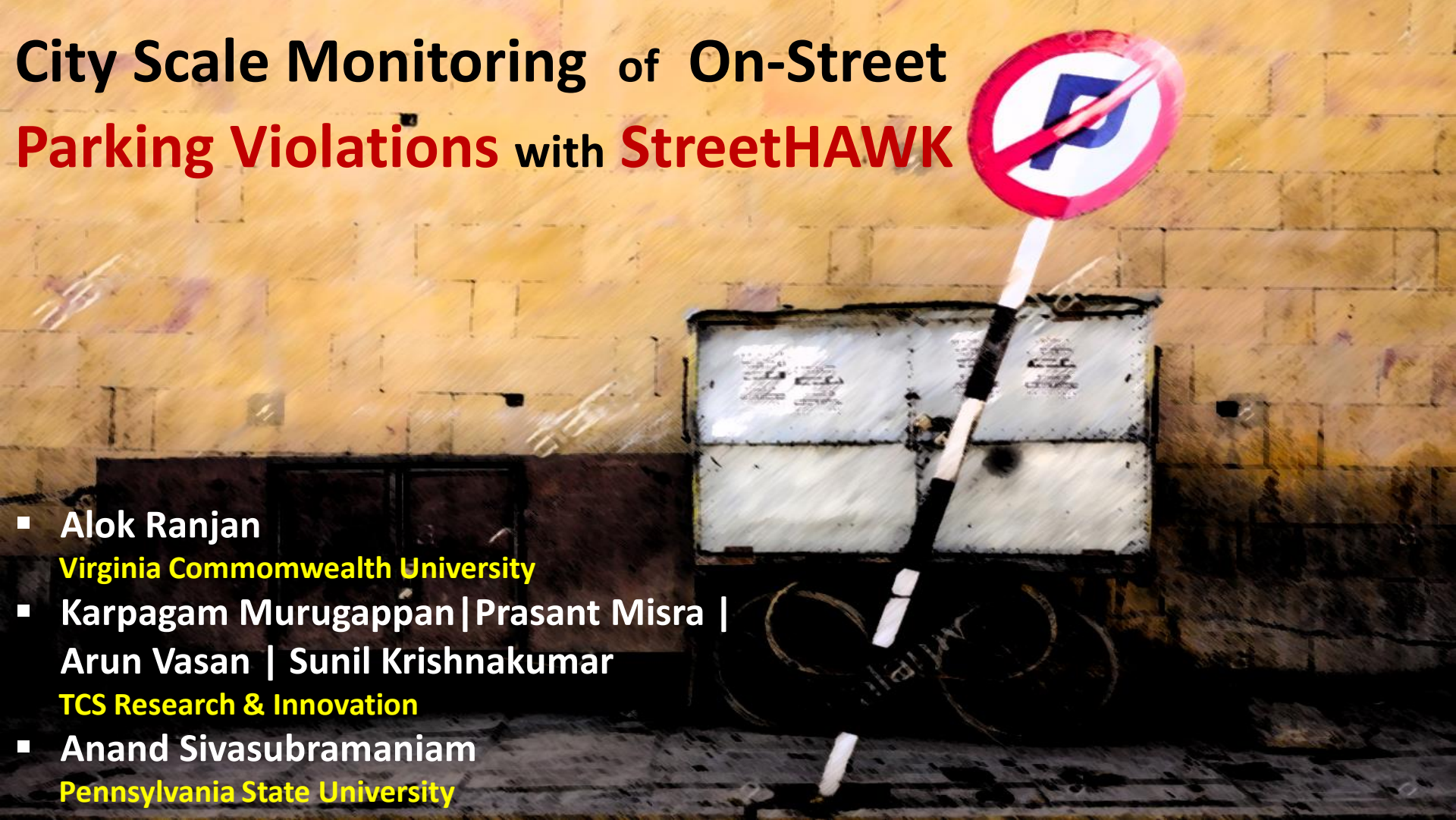
City Scale Monitoring of On-Street Parking Violations with StreetHAWK



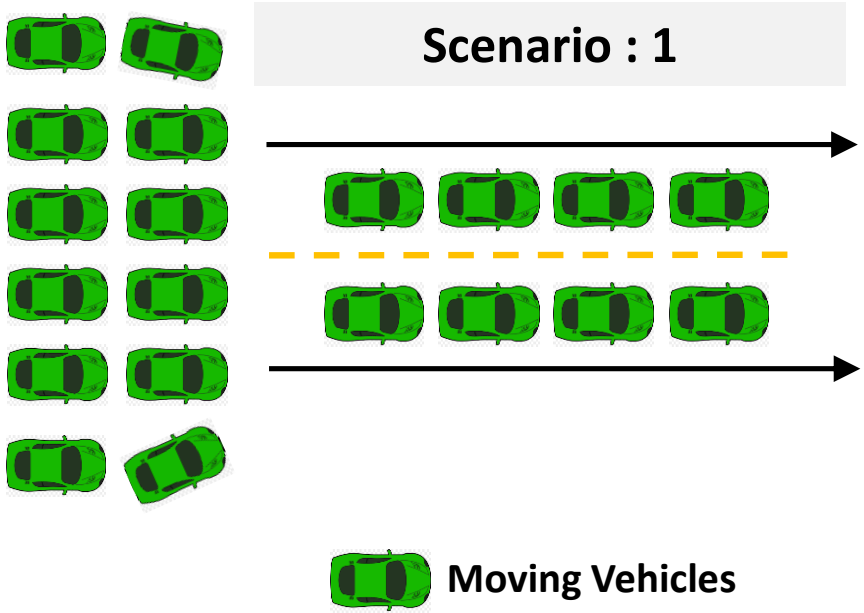
- Alok Ranjan
  Virginia Commomwealth University
- Karpagam Murugappan|Prasant Misra | Arun Vasan | Sunil Krishnakumar
  TCS Research & Innovation
- Anand Sivasubramaniam
  Pennsylvania State University

Slow Moving Traffic | Traffic Congestion | Developing Countries

Traffic Congestion and On-street Parking Violations?

TCS RESEARCH & INNOVATION

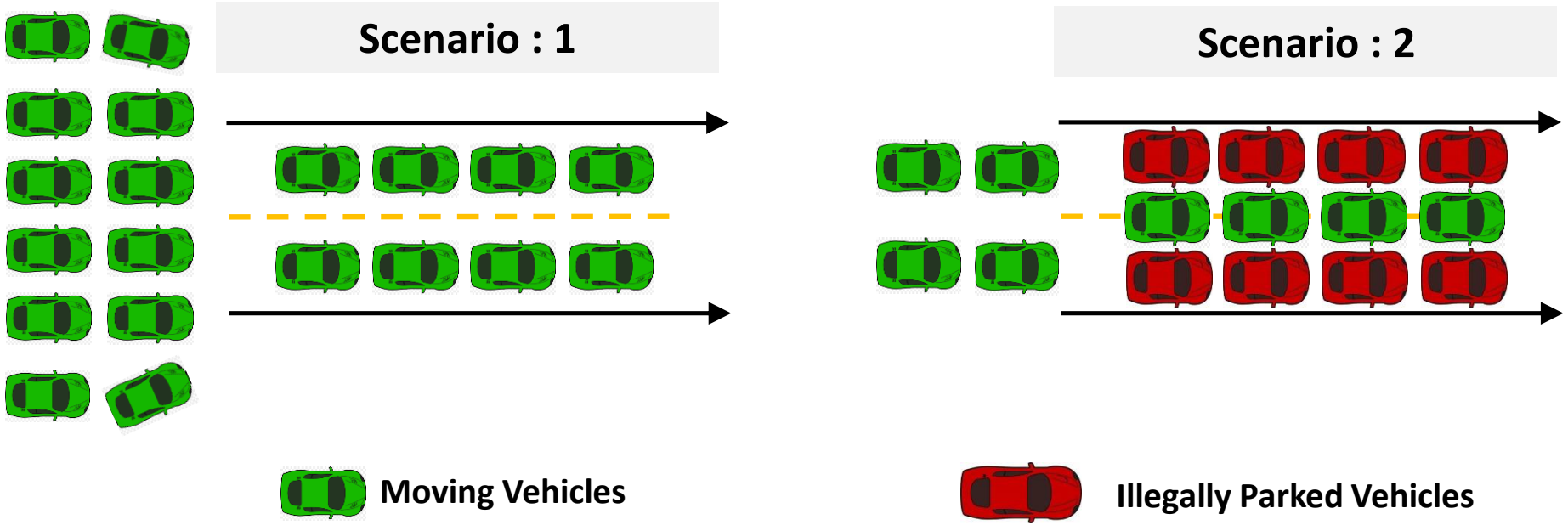
Congestion := Demand exceeds Capacity



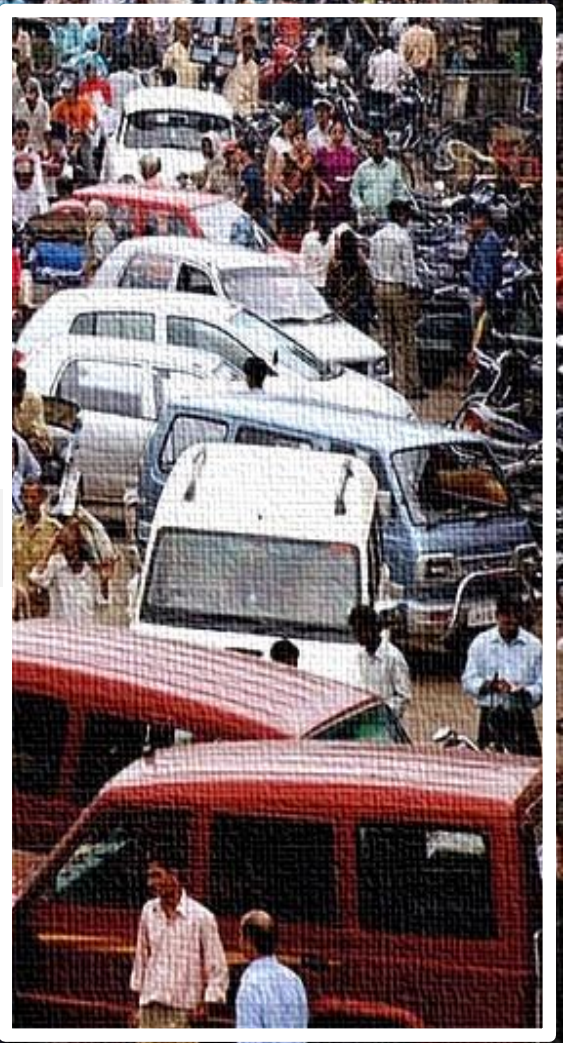
Traffic Congestion and On-street Parking Violations?

TCS RESEARCH & INNOVATION

Congestion := Demand exceeds Capacity



5



Illegal parking takes up 40% of road/street space!

Problem Statement

How to **monitor** **illegal** **parking** practices at **city scale** with **limited** human intervention?

State of Current Practice

TCS RESEARCH & INNOVATION

How is it done now?

- Focus is on identification and deterrence!
- Identification best practices
  - (Semi) manual analysis of video feeds from traffic surveillance cameras
  - Patrolling by enforcement agencies
- Deterrence procedures
  - Issue of violation tickets with monetary fines; towing; wheel clamping

Limitations of current practice?

- Monitoring is (mostly) manual
- Difficult to scale-up and maintain the monitoring system with problem size; and is costly
- Streaming visual data to the cloud has privacy implications

The StreetHAWK Proposal

Piggyback on the ubiquitous smartphone infrastructure that is readily available on radio taxis moving in the city!

The StreetHAWK System

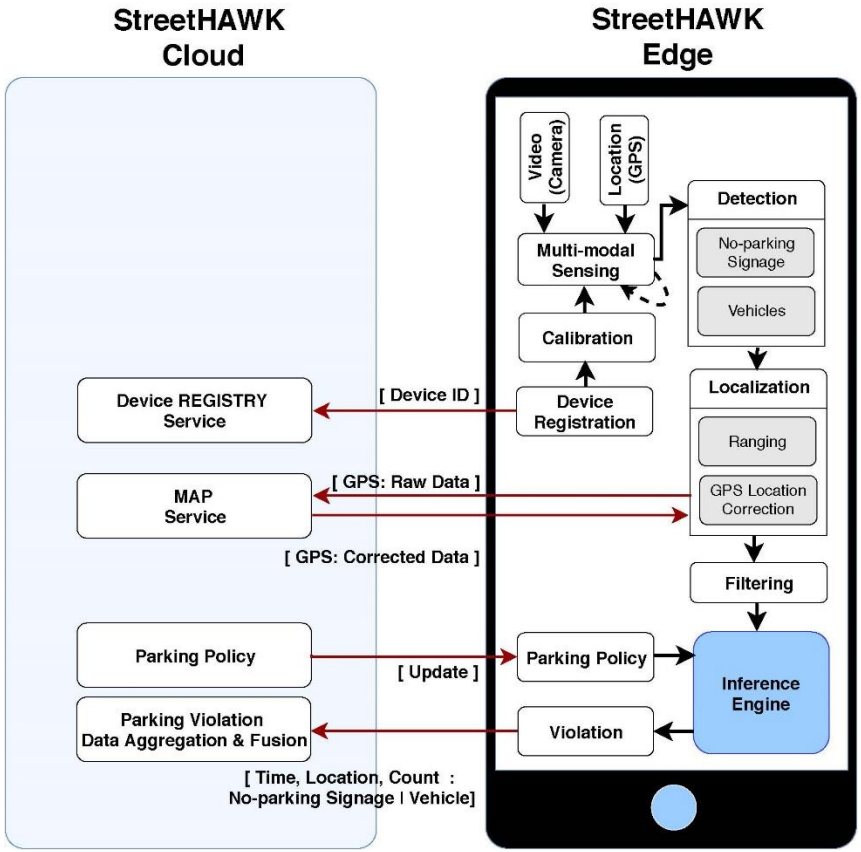
Scope of Operation

- View the roadside parking scene using the rear camera of the dashboard/windshield mounted smartphone
- Locally process the video feed in real-time to identify no-parking signages and vehicles parked in its vicinity
- Match it with parking policies to detect the location of illegally parked vehicles
- Send the status report to the cloud that can further notify various city agencies

Features

- Easy to Scale
  - City taxi networks are a widespread mode of transport | Egde based system
- Low in Cost and Deployment Complexity
  - Makes use of off-the-shelf smartphones that are already present in taxis
  - Precludes the need for additional instrumentation, either in urban environment or on the vehicle
- Real-time and Privacy Preserving
  - Performs on-board analytics on the phone itself, without sending back raw video data to the cloud
  - The status report is sent to the cloud, which consists of the location of the illegally parked vehicles
- Automated
  - On-time installation of the StreetHAWK app on the smartphone and granting it the correct system access privileges is ALL that is needed from the user point of view. The rest of the workflow is automated.
- Non-disruptive to the normal process

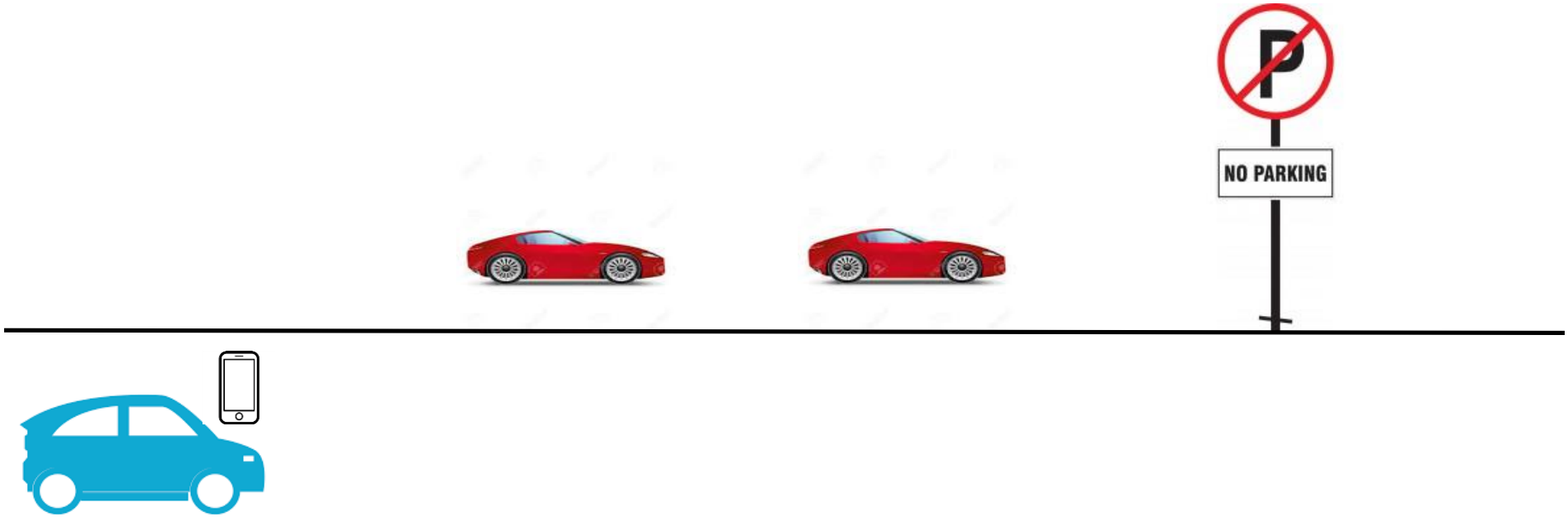
StreetHAWK: High-level Architecture



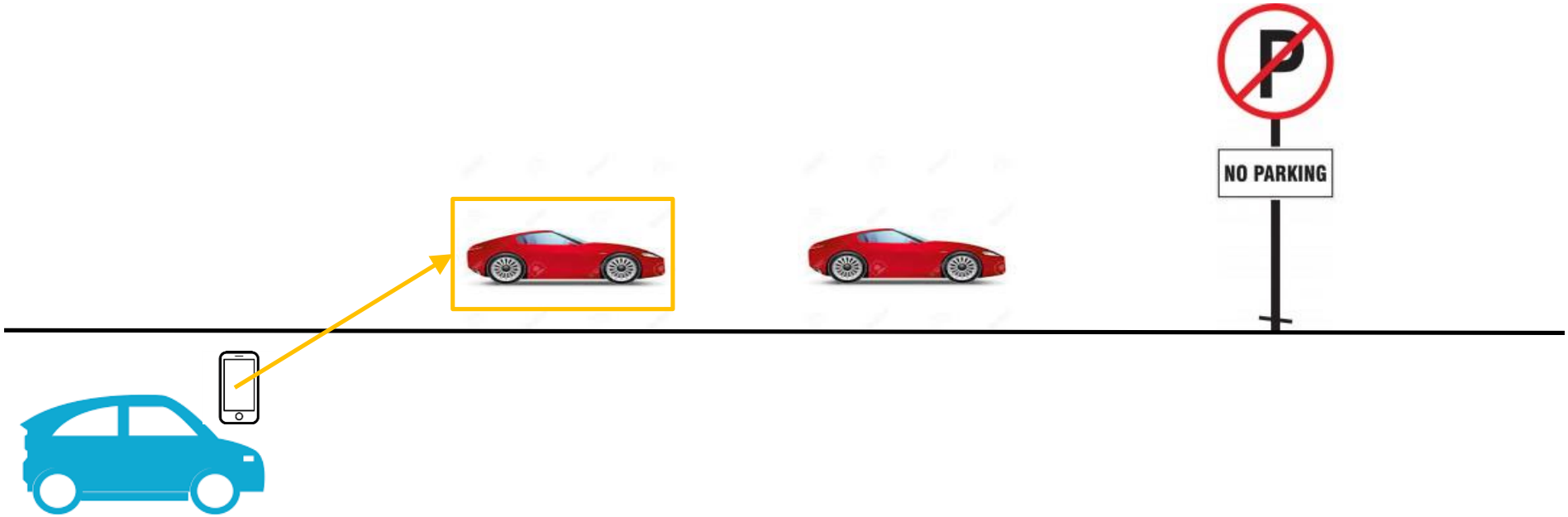
System level Challenges

- High-speed mobile vision with an inverted monitoring architecture
- "Zero" control over the set-up
- Edge platforms are constrained (in comparison to the cloud infra)
- Smartphone based edge system offer a shared working environment | cannot get greedy about platform resources

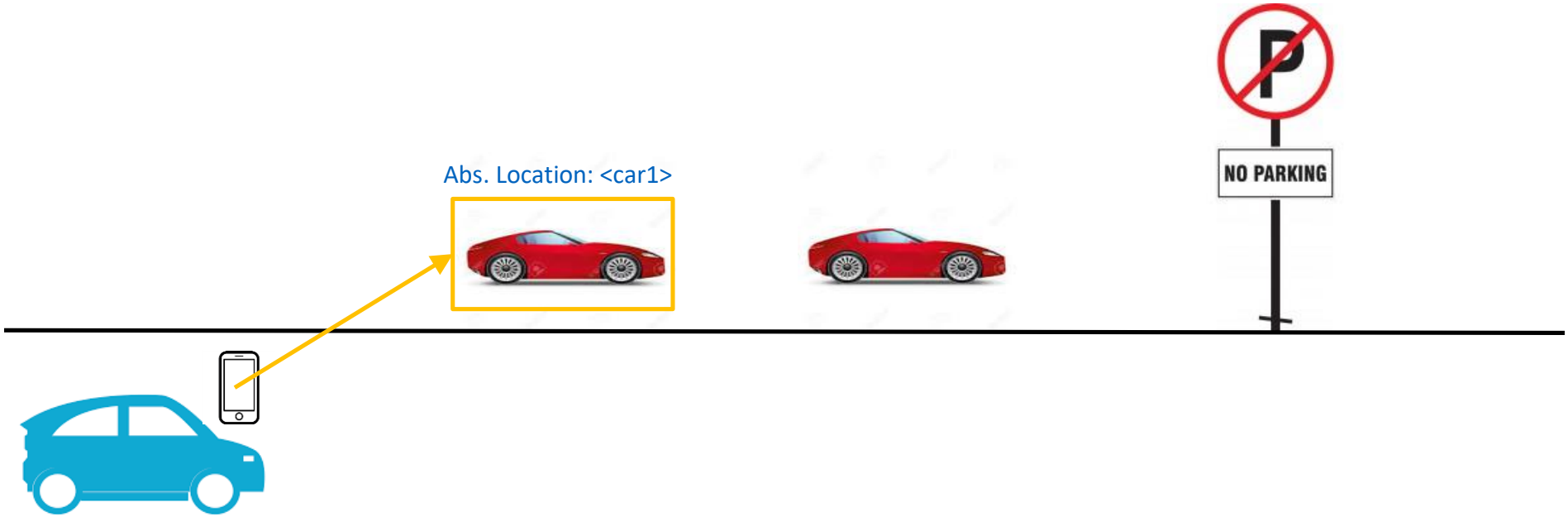
StreetHAWK: Functional Flow



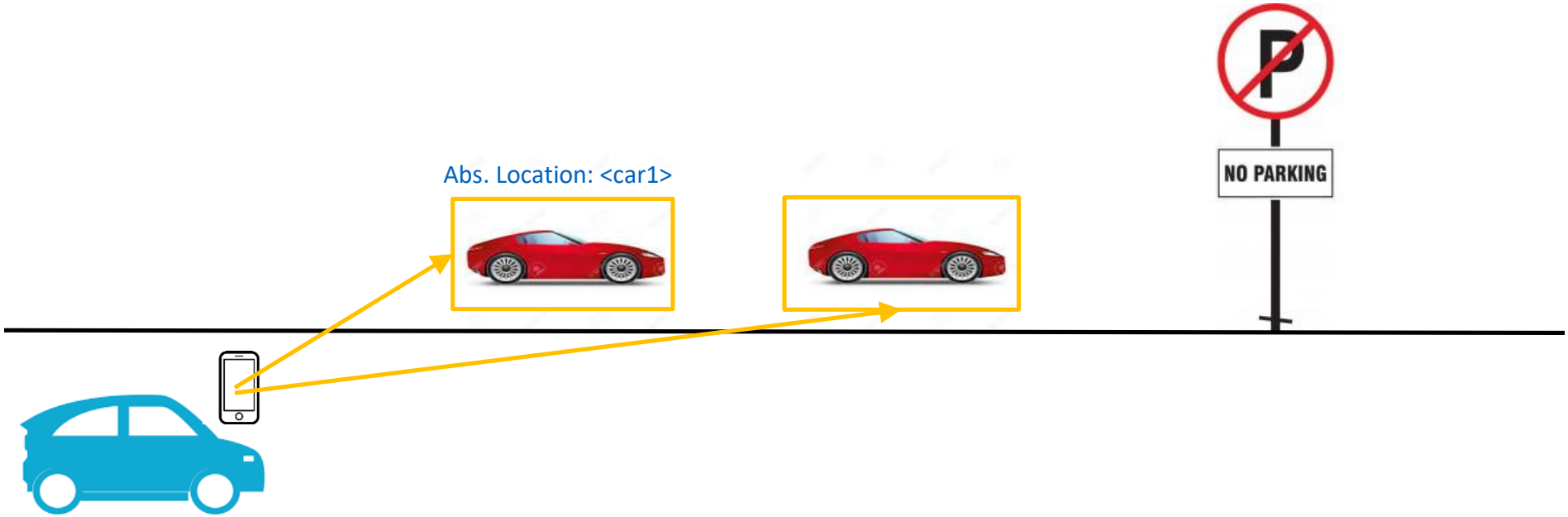
StreetHAWK: Functional Flow



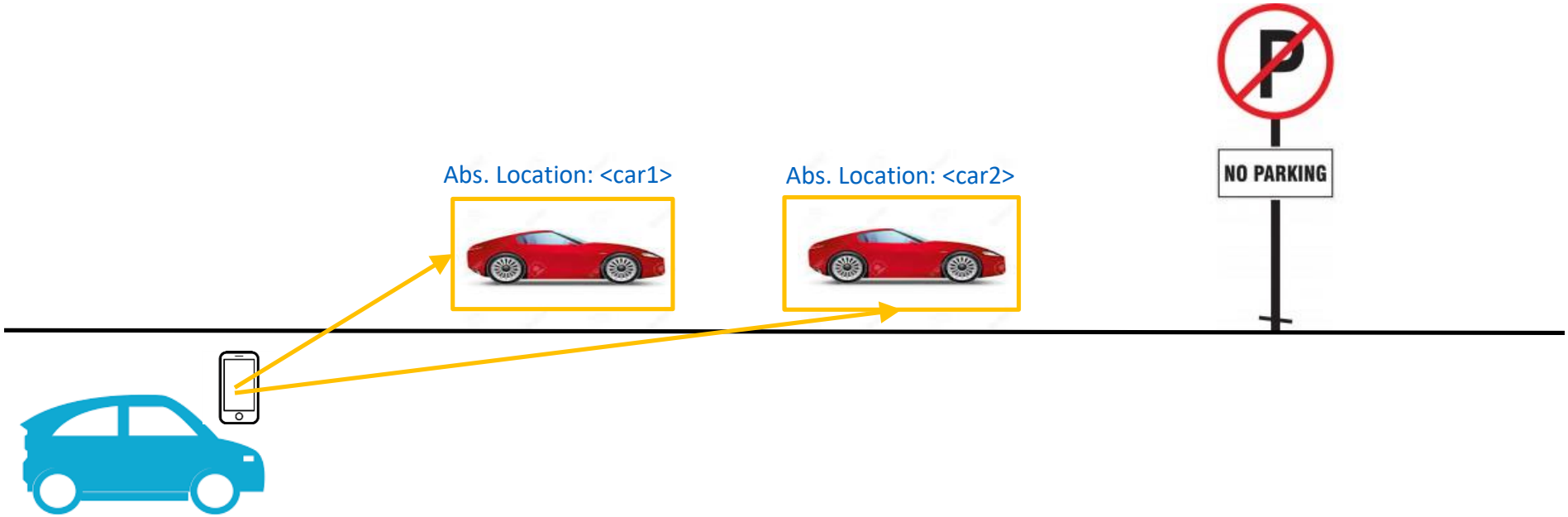
StreetHAWK: Functional Flow



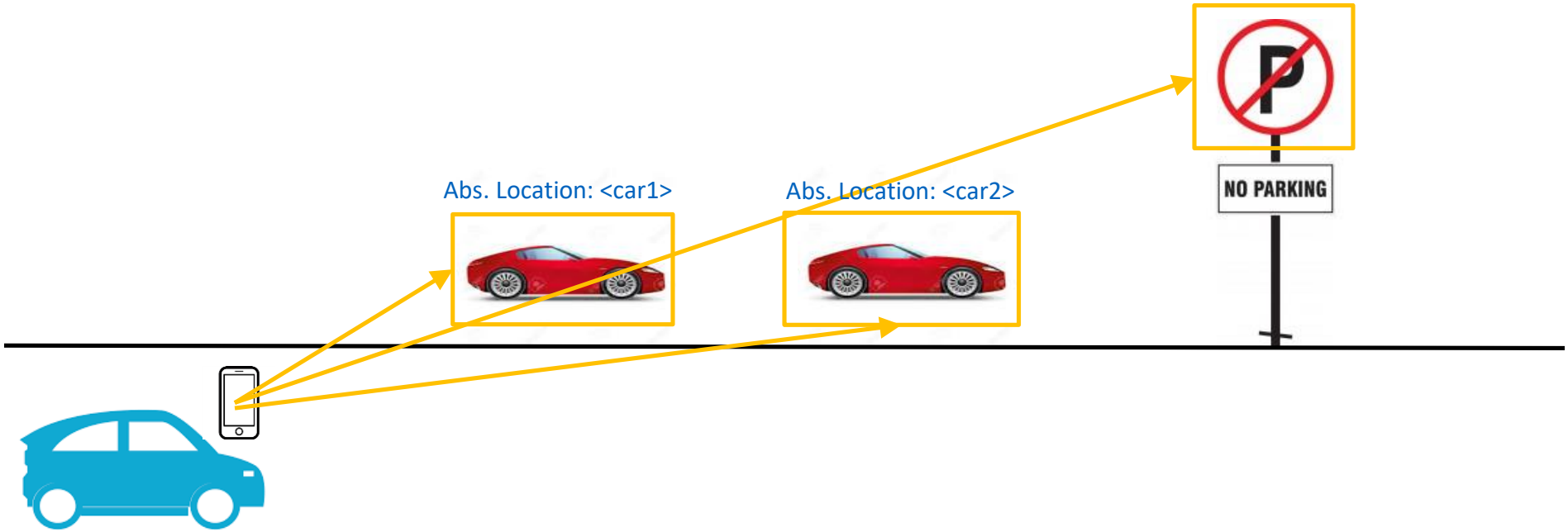
StreetHAWK: Functional Flow



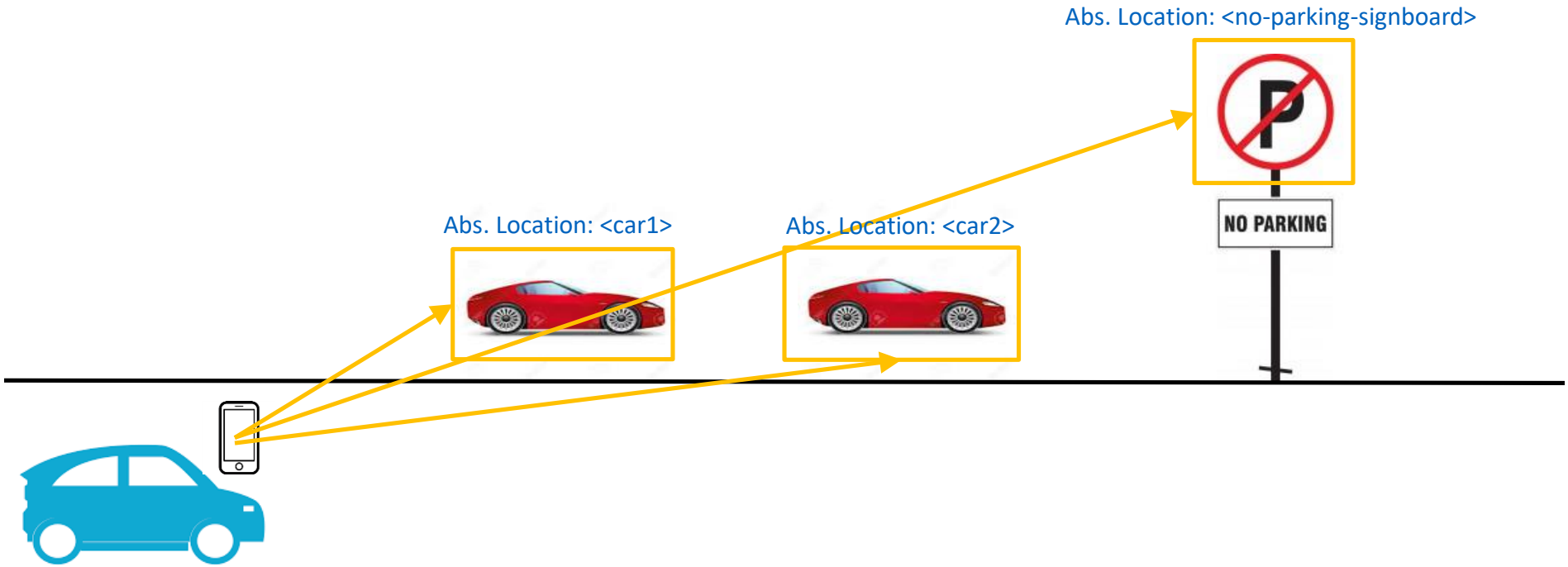
StreetHAWK: Functional Flow



StreetHAWK: Functional Flow



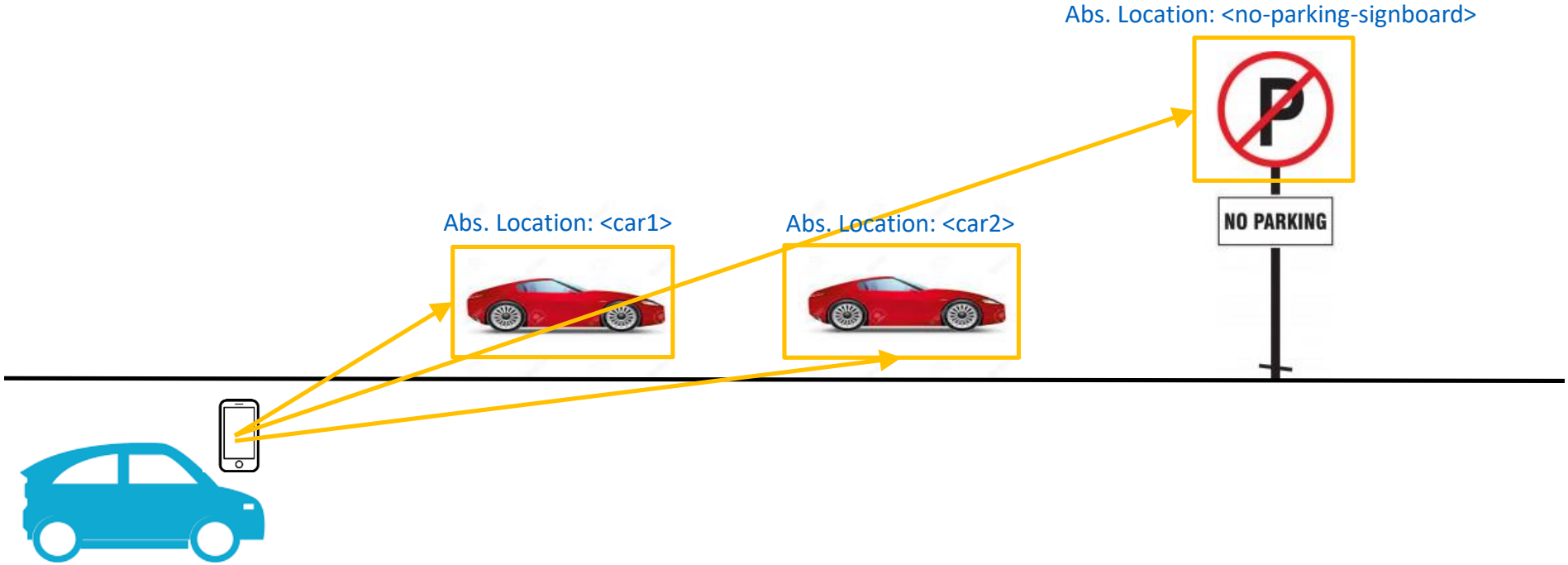
StreetHAWK: Functional Flow



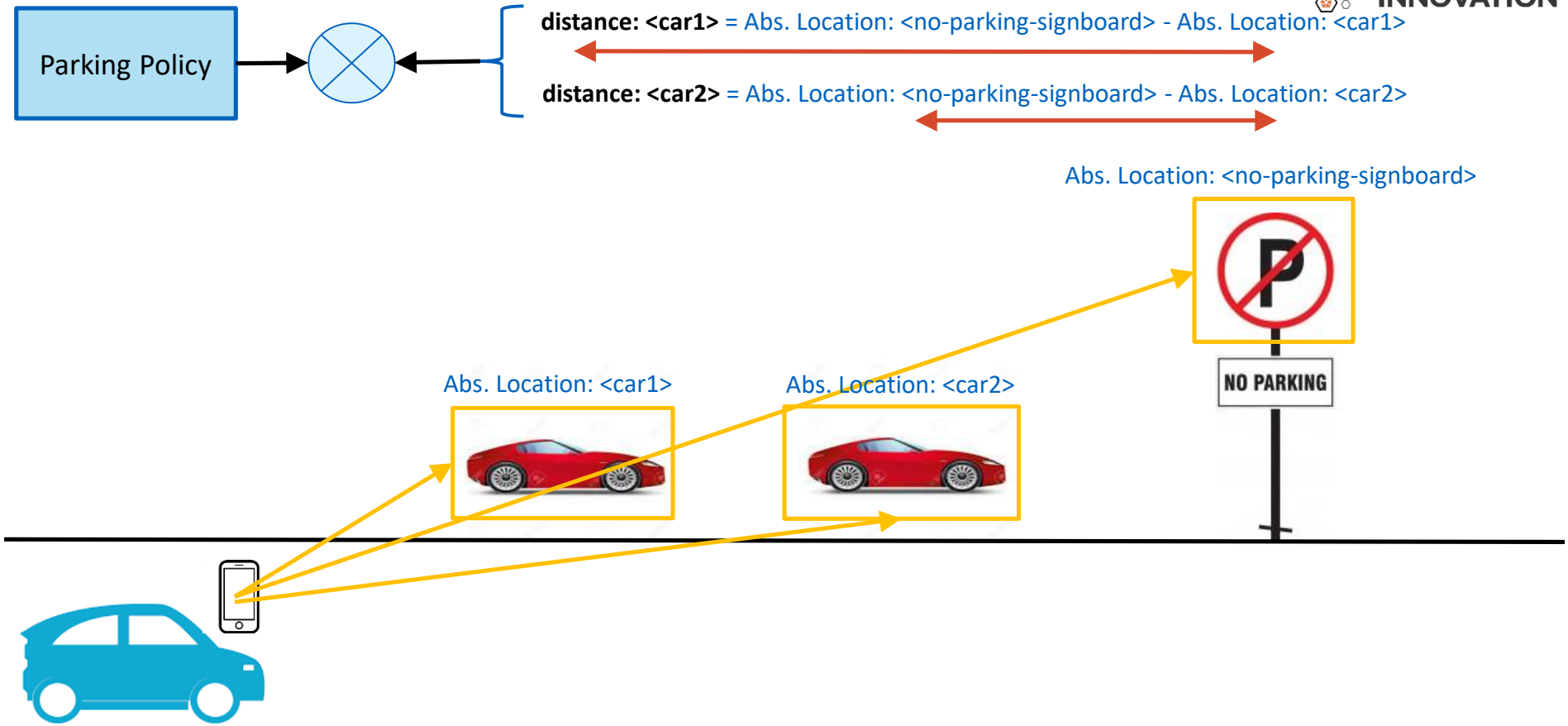
StreetHAWK: Functional Flow

distance: <car1> = Abs. Location: <no-parking-signboard> - Abs. Location: <car1>

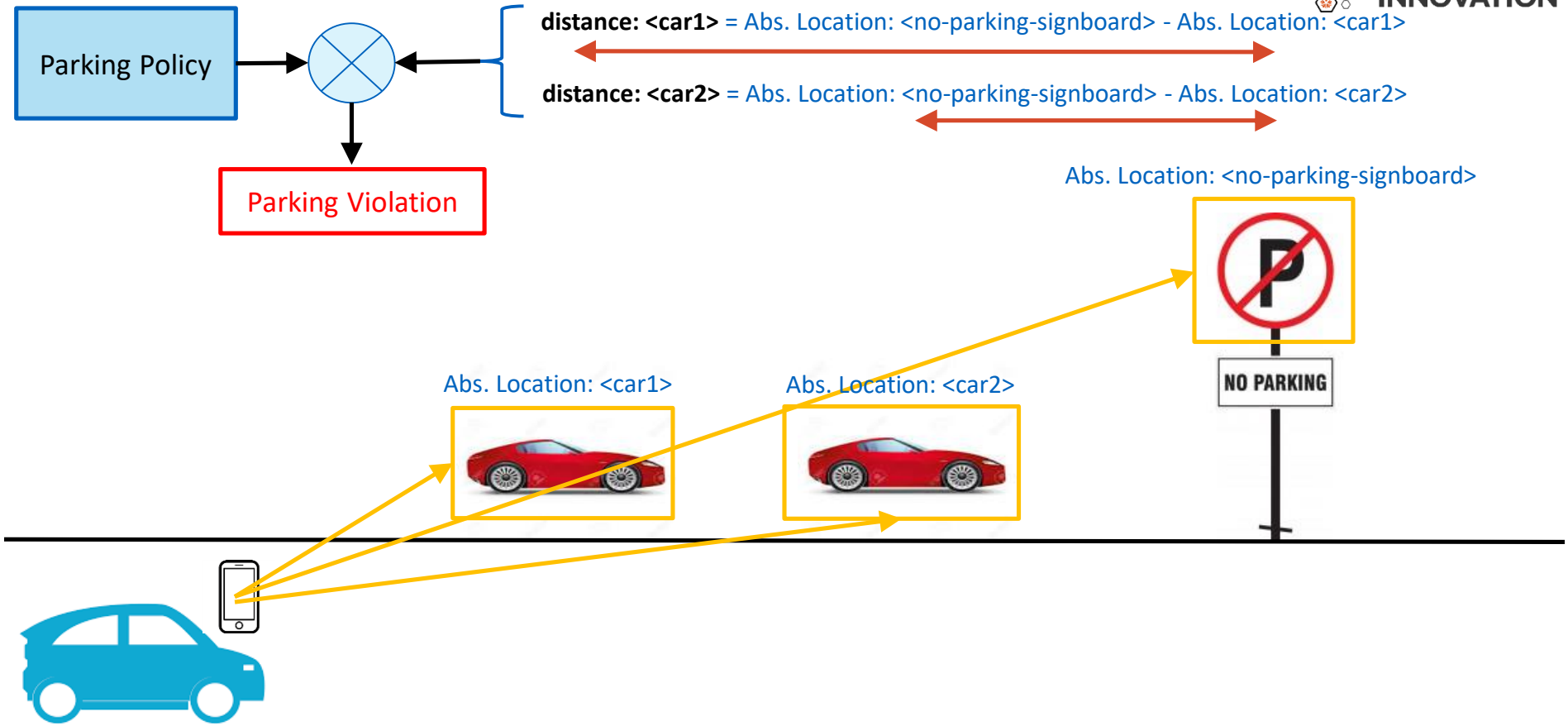
distance: <car2> = Abs. Location: <no-parking-signboard> - Abs. Location: <car2>



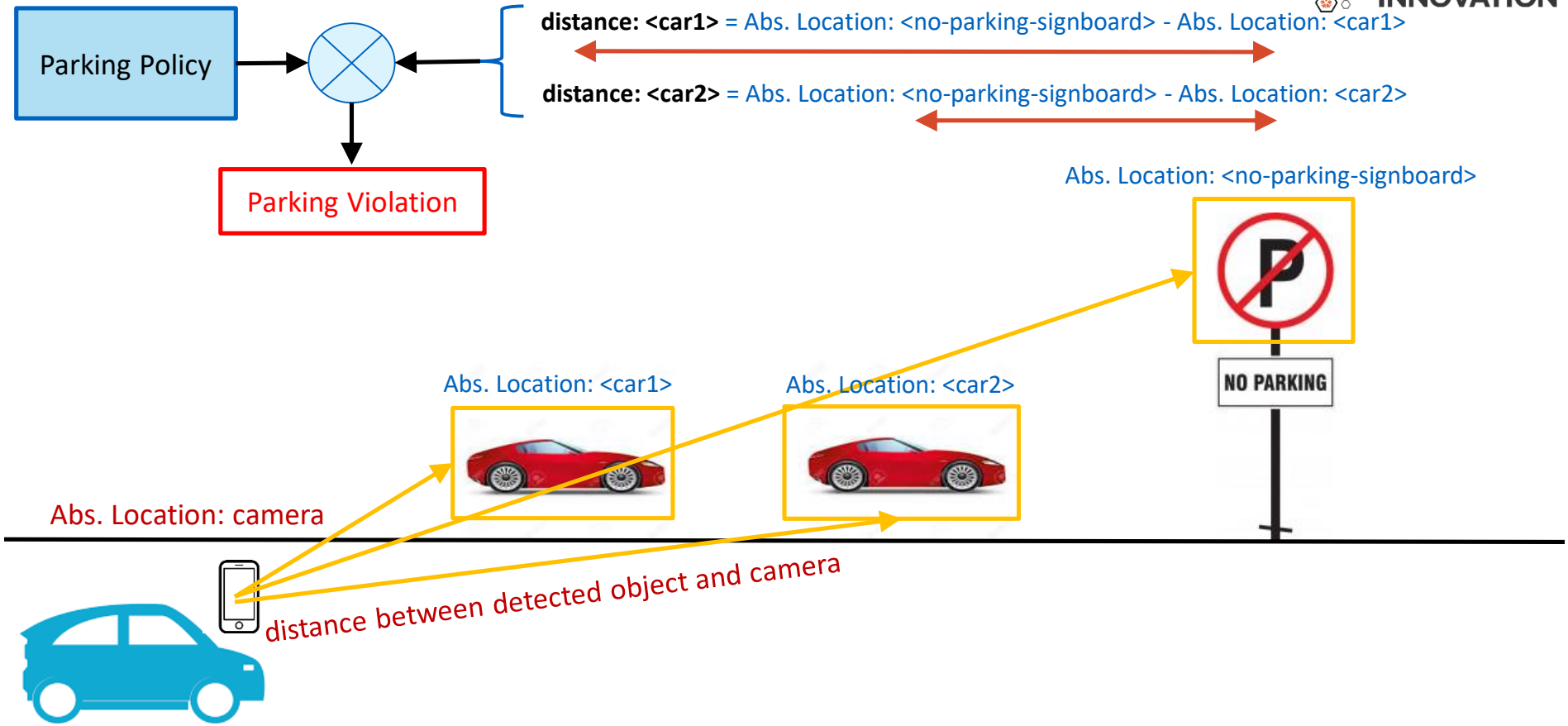
StreetHAWK: Functional Flow



StreetHAWK: Functional Flow



StreetHAWK: Functional Flow



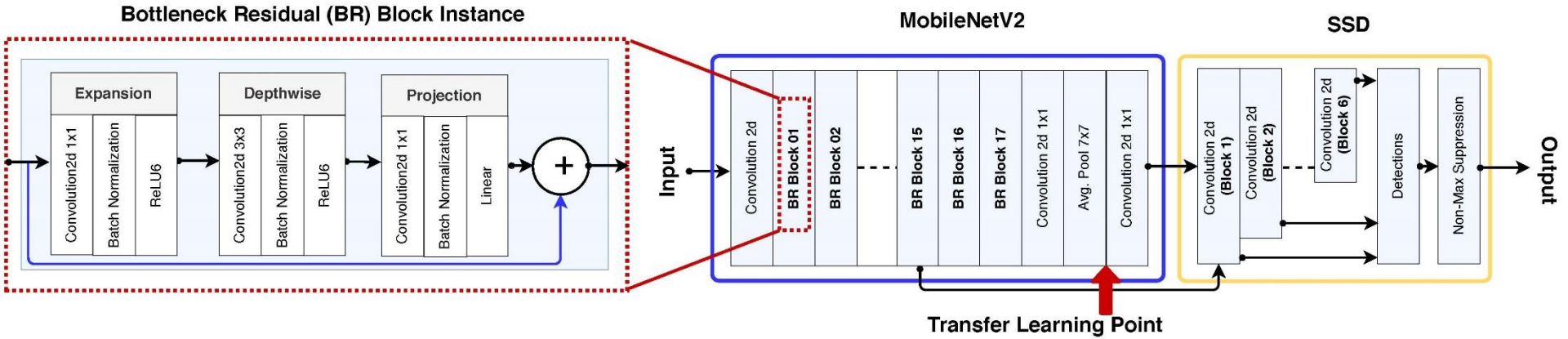
StreetHAWK: Object Detection Challenges

- **Aim**: Detect objects-of-interest (no-parking signboards | motorbikes | auto-rickshaws | cars)

- **Challenges**:
  1. Small object detection
  2. Lack of consistent pattern for visual detection (*more applicable to developing countries*)
     - Irregularities in the deployment of no-parking signboards
     - Non-standard manner of vehicle parking
  3. Unique object identification
     - Multiple detection of the same object across frames
  4. Differentiate : moving vs. parked vehicle
     - Recall: system uses a mobile setup | camera is moving | scene is moving

StreetHAWK: Object Detection Model

Convolutional detection model : SSD meta-architecture with MobileNetV2 as the feature extractor

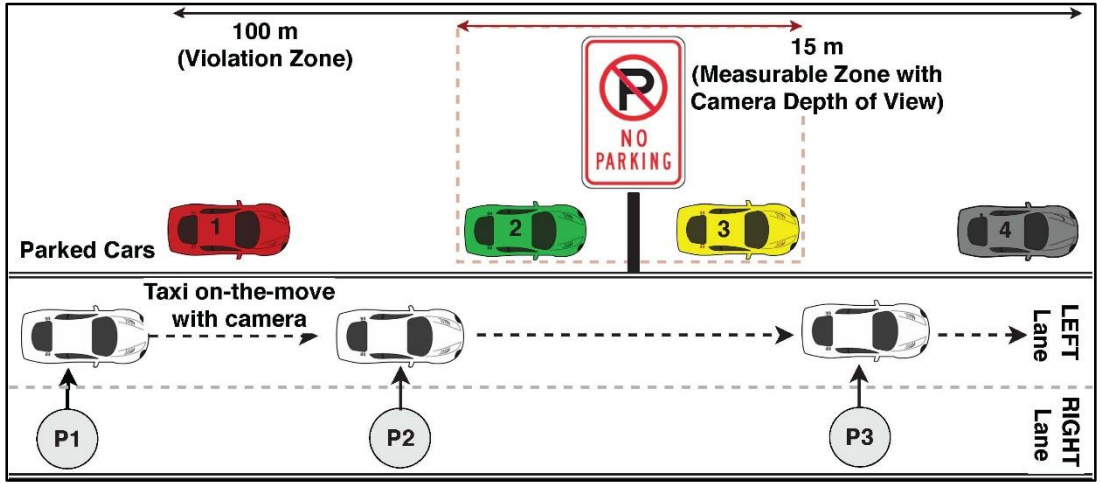


- **Approach**
  - Used a detection model pre-trained on the MS COCO dataset
  - Limited object categories to 4!
  - Derived a new model by using transfer learning and re-training with custom dataset collected under a wide range of real-world constraints and environmental conditions (with data augmentation)

Overcomes object detection challenges 1 and 2 !

StreetHAWK: Ranging Challenges

- **Aim**: Find the distance between the camera and the detected objects
- **Challenges**:
  - Since the camera is moving, there is a need for an ultra-fast visual distance measurement technique
  - Limited measurement range with a single camera system
    - The single shot detection range is limited to 15m, while the field requirement is that of 100m



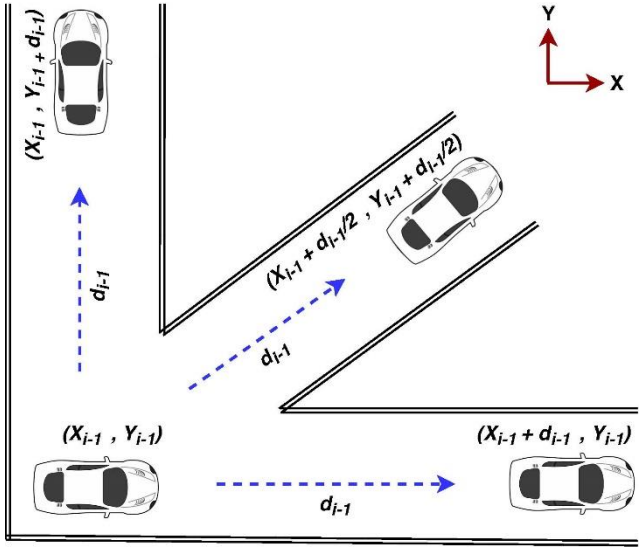
- **Approach**:
  - Used the pinhole principle to measure the distance between the camera and the detected objects
  - Used a short-term historian to successively log the details of the detected objects

StreetHAWK: Localization Challenges

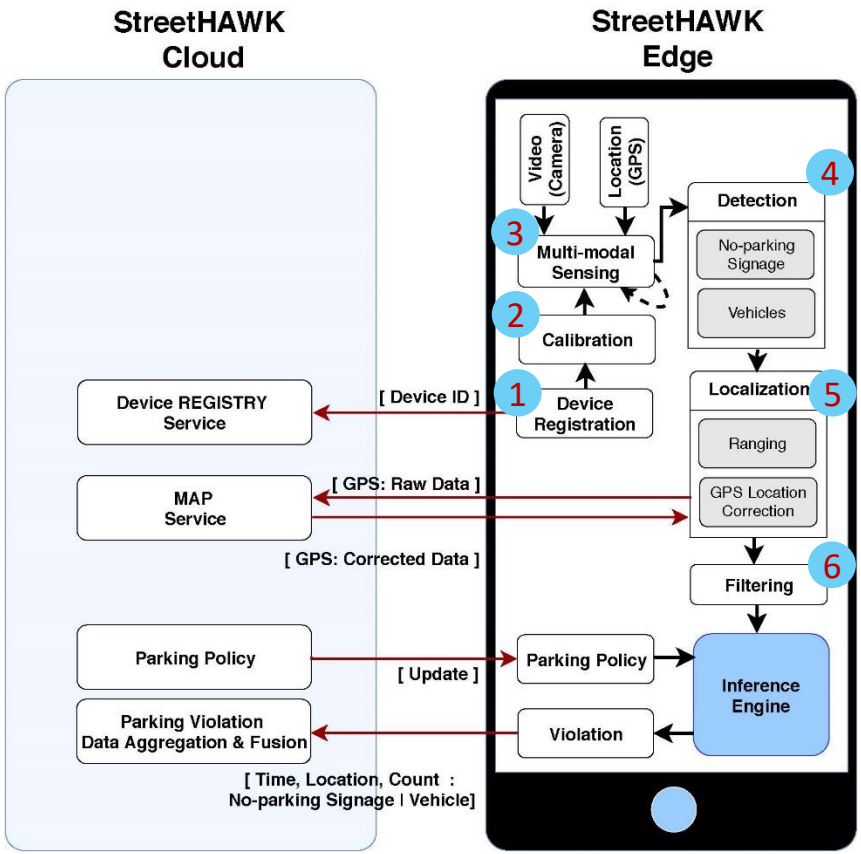
- **Aim**: Find the absolute location of the detected objects

- **Challenges**:
  - For real-time operation:
    - localization needs to be performed on the device itself
    - localization technique must be lightweight

- **Approach**:
  Find out if <x, y, or both x and y> co-ordinates of the camera location are changing
  - Wait for at least 2 consecutive object detections
  - If <x> co-ordinate is changing, then add the Distance (camera->object) to x!
  - If <y> co-ordinate is changing, then add the Distance (camera->object) to y!
  - If both <x and y> co-ordinates are changing, then add Distance (camera ->object)/2 to both <x> and <y>



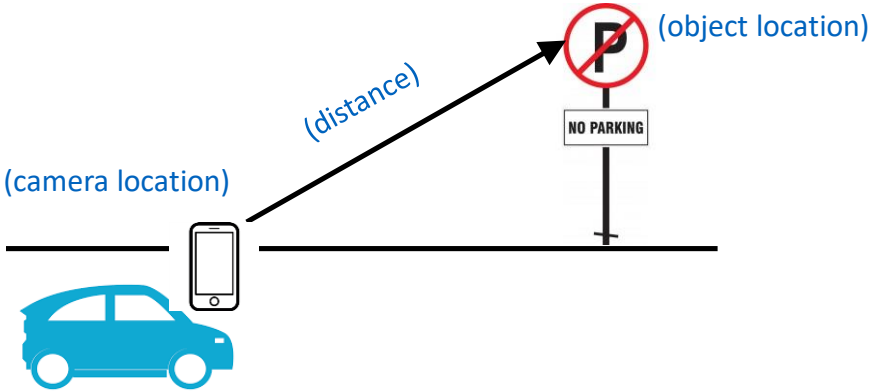
StreetHAWK: Functional Flow



Historian

Record (for EACH detected object):

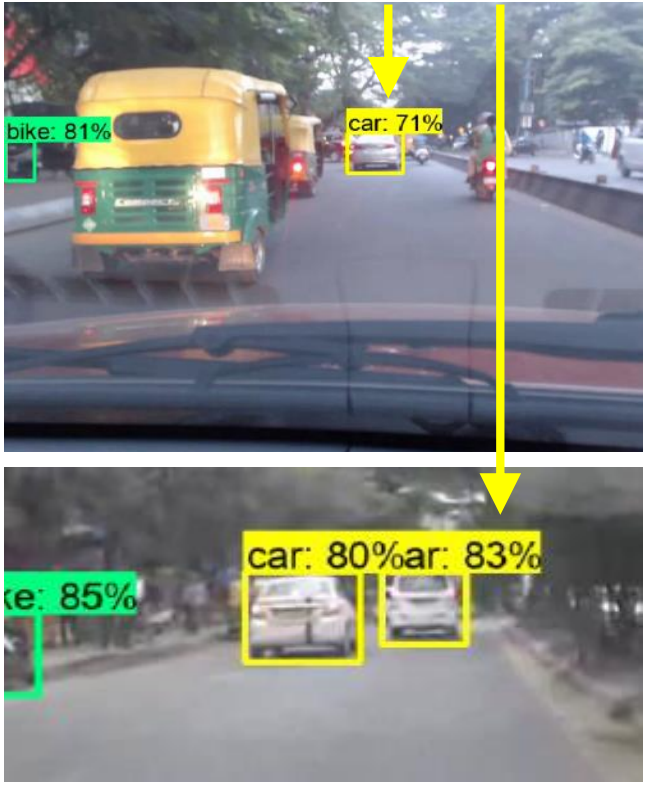
- (Time) of detection
- Camera (location)
- (Distance) between camera and detected object
- Object (location)



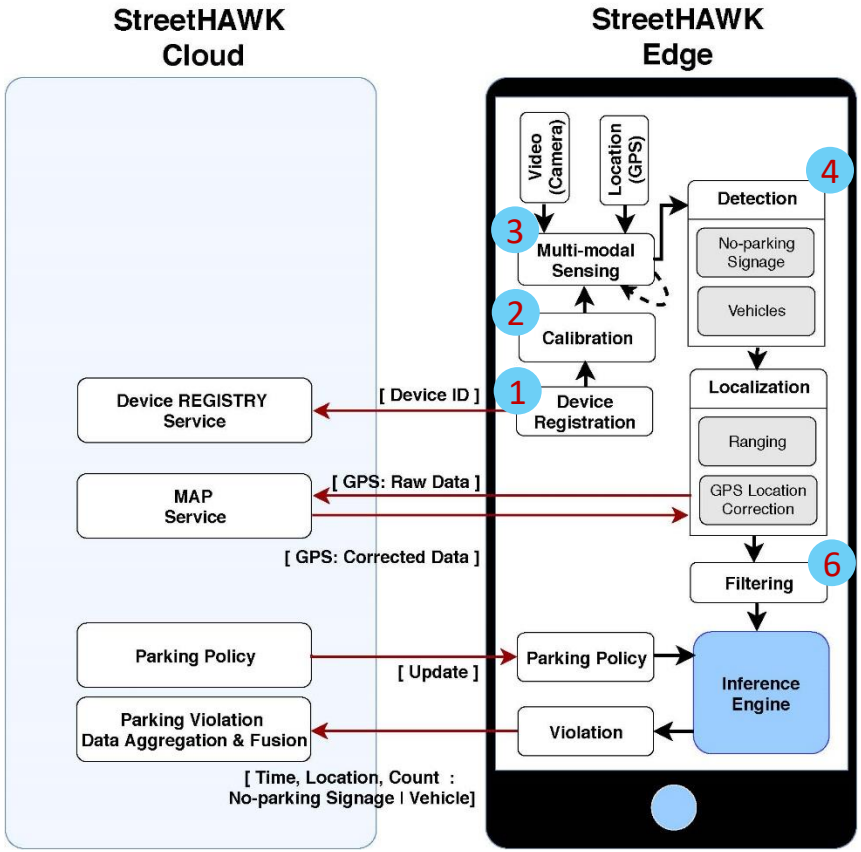
StreetHAWK: Filtering

- **Aim**: remove erroneous object detections
- **Challenges**:
  1. Unique object identification
     - Multiple detection of the same object across frames
  2. Differentiate : moving vs. parked vehicle
     - Recall: system uses a mobile setup | camera is moving | scene is moving
- **Approach**:
  - Unique object identification
    - Perform sequential clustering of object locations
      - Calculate the distance between two consecutive object locations
      - If distance < 1m, then the object locations are clustered together as belonging to the same detected object
  - Differentiate between a parked and a moving vehicle
    - If the camera is moving, and as the camera gets closer to the detected vehicle:
      - If the distance between the camera and the detected vehicle decreases, then detected vehicle = PARKED
      - For anything else, the detected vehicle is moving

Moving vehicles are detected as PARKED



StreetHAWK: Functional Flow @ Inferencing

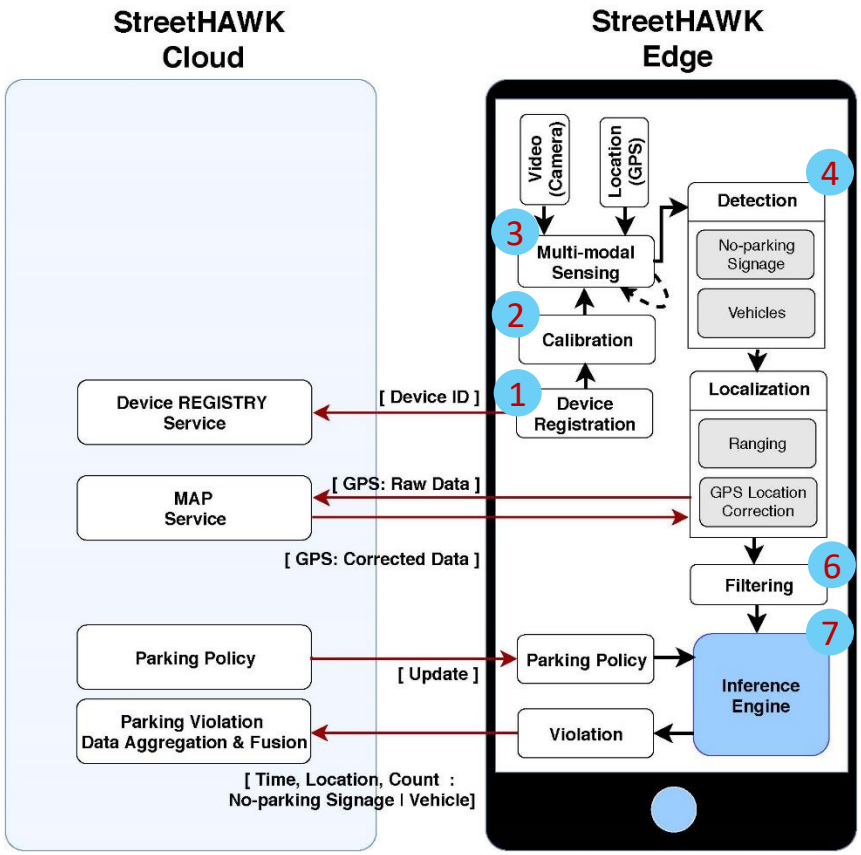


Historian

For each detected object:

- (Time) of detection
- Camera (location)
- (Distance) between camera and detected object
- Object (location)

StreetHAWK: Functional Flow @ Inferencing



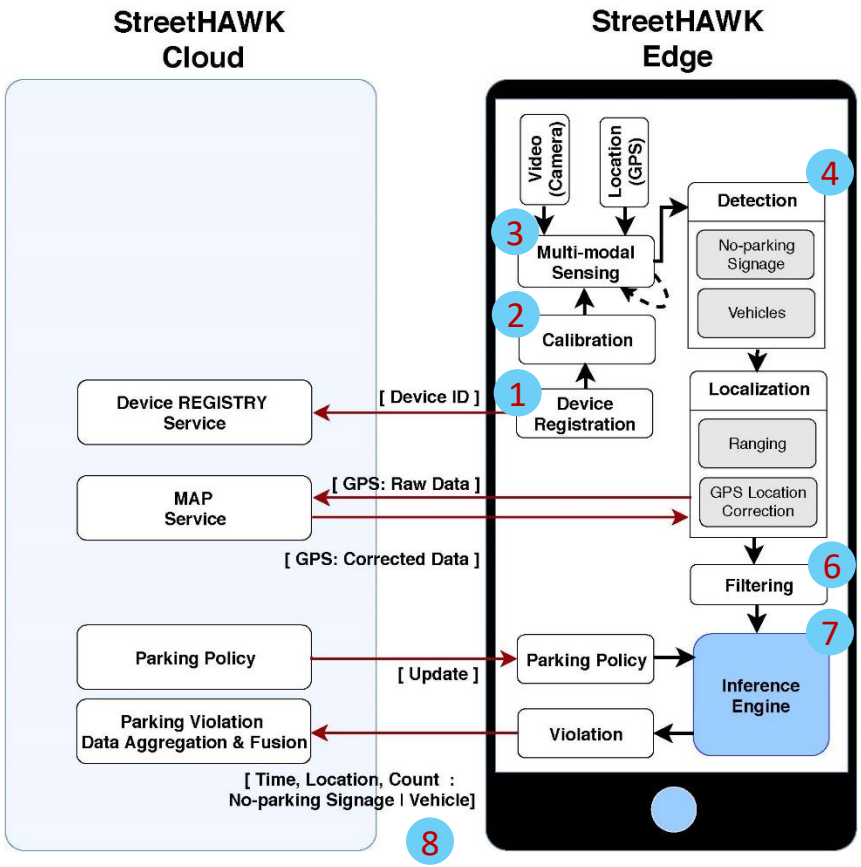
Historian

For each detected object:

- (Time) of detection
- Camera (location)
- (Distance) between camera and detected object
- Object (location)

- Trigger the inference engine when a no-parking signage is detected
- Calculate the distance of all parked vehicles to the no-parking signage (refer to HISTORIAN)
- Based on parking policy, flag VIOLATIONS!

StreetHAWK: Functional Flow @ Inferencing



Historian

For each detected object:

- (Time) of detection
- Camera (location)
- (Distance) between camera and detected object
- Object (location)

- Trigger the inference engine when a no-parking signage is detected
- Calculate the distance of all parked vehicles to the no-parking signage (refer to HISTORIAN)
- Based on parking policy, flag VIOLATIONS!

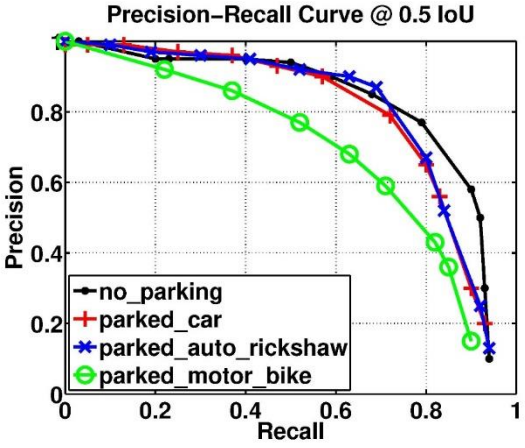
StreetHAWK: Evaluation Results

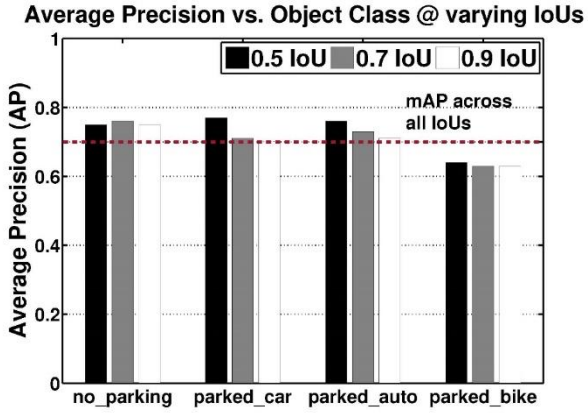
- We collected and manually labeled a dataset extracted from over 50 hours of citywide driving video feeds that spanned over 4 months
- We performed on-the-road experiments that spanned close to 500 km

StreetHAWK: Object Detection Performance

TCS RESEARCH & INNOVATION

PASCAL VOC Evaluation Benchmarks



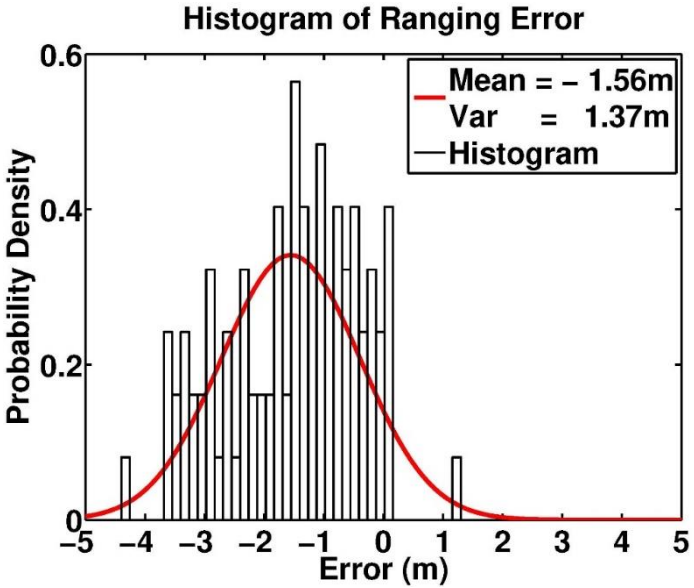


MS COCO Evaluation Benchmarks

Method	Category	mAP, IoU : (0.50 - 0.95)	mAP, IoU : (0.50)	mAP, IoU : (0.75)	mAP, Area : (S)	mAP, Area : (M)	mAP, Area : (L)	mAR, maxDets : (01)	mAR, maxDets : (10)	mAR, maxDets : (100)	mAR, Area : (S)	mAR, Area : (M)	mAR, Area : (L)
MobileNetV1 + Faster RCNN	Embedded	0.17	0.41	0.10	0.01	0.17	0.28	0.21	0.30	0.31	0.08	0.32	0.51
MobileNetV1 + SSD300	Embedded	0.22	0.48	0.18	0.05	0.24	0.29	0.28	0.34	0.34	0.11	0.34	0.45
VGCNet + SSD300	Non-Embedded	0.23	0.41	0.23	0.05	0.23	0.41	0.23	0.33	0.35	0.10	0.38	0.57
MobileNetV2 + SSD300*	Embedded	**0.26**	**0.55**	**0.21**	**0.07**	**0.28**	**0.37**	**0.32**	**0.42**	**0.43**	**0.20**	**0.45**	**0.58**

StreetHAWK: Ranging Performance

Static Ranging



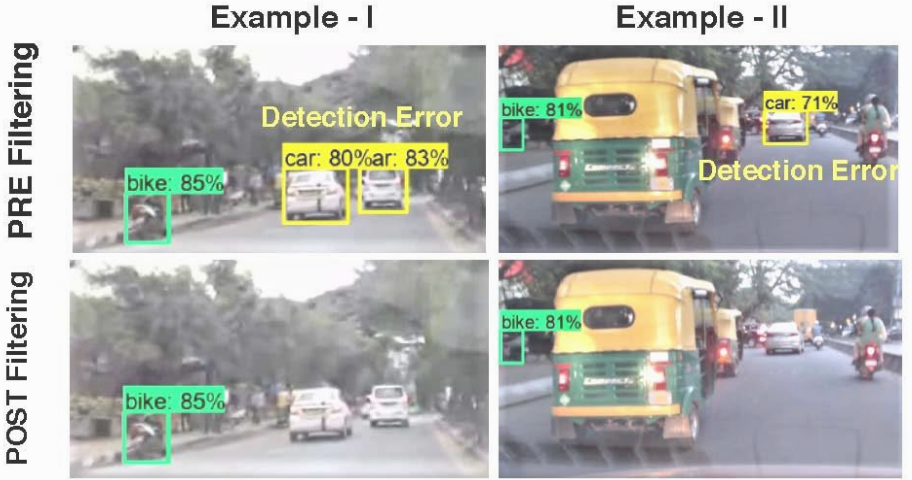
Ranging error under stationary condition, where it is observed to be less than 4 m.

Mobile Ranging

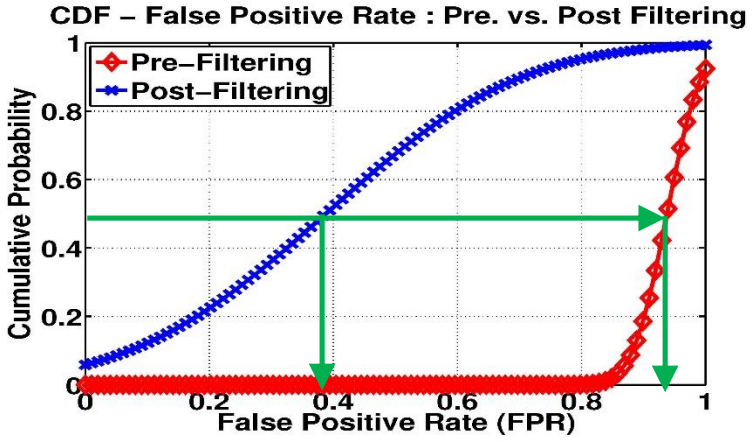
Distance (m)	Error (m), Angle :		
	$05^O \rightarrow 20^O$	$20^O \rightarrow 30^O$	$30^O \rightarrow 60^O$
01 ⟶ 05	–	–	(-) 2-3
05 ⟶ 10	–	(-) 3-5	–
10 ⟶ 15	(-) 3-5	–	–

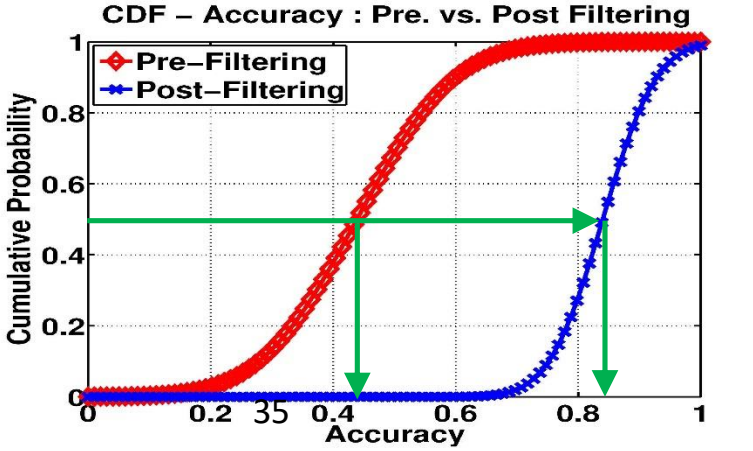
Ranging error under mobile condition; where the maximum error is observed to be less than 5m across all cases.

StreetHAWK: Filtering Performance



Representative scenarios where moving vehicles are incorrectly detected as parked, and removal of those errors post filtering.





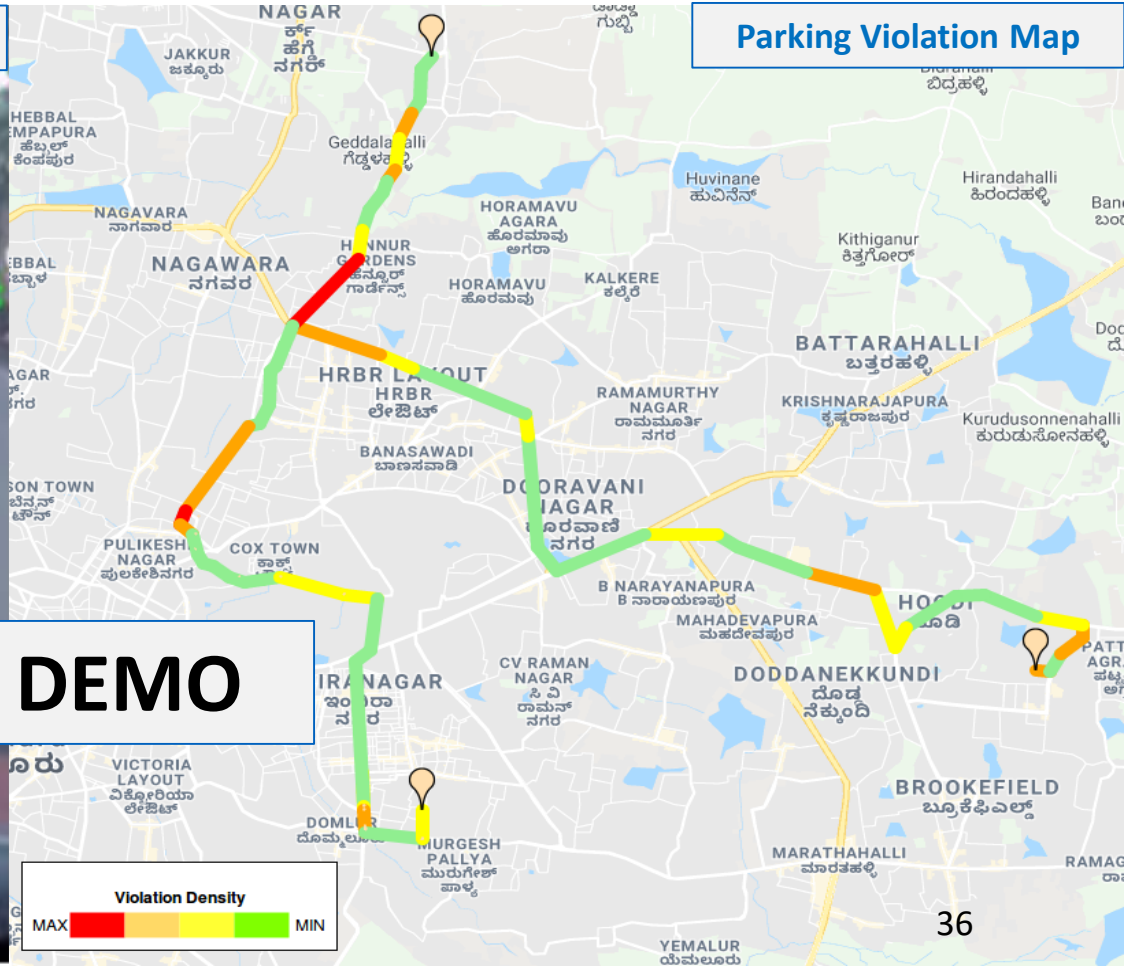
Filtering techniques significantly decrease the false positive rate; as a result of which, the end-to-end system accuracy increases by a factor of two.

StreetHAWK: On-the-road Trial

TCS RESEARCH & INNOVATION

StreetHAWK UI

Parking Violation Map



DEMO

Conclusion

- With an end-to-end design and implementation of StreetHAWK for parking violation monitoring, we demonstrate the feasibility of a COTS smartphone based edge system!
  - combines a single camera visual sensing mode with edge-compatible machine learning and analytic models.

- We account for the edge platform constraints and propose lightweight methods for detecting parking violations and measuring the violation span/density.
  - We use a deep neural network (DNN) based convolution detection model, and address the model limitation of identifying small objects in a wide variety of real-world conditions by extensive training and parameter tuning.
  - We use the visual ranging capability of a single camera phone system to measure the violation span, but enhance it with a short-term historian and GPS to extend the system range from 15m to the prescribed 100 m.
  - At the overall system level, we make use of the mobility of the camera unit and multi-modal sensing clues to filter out erroneous violation instances.

- System performance
  - three times better accuracy in detecting small sized objects than other state-of-the-art embedded models
  - worst case ranging error of less than 5 m
  - operates at a speed of 5 frames per second (FPS) on typical mid-range Android smartphones
  - identifies, on an average, 80% of parking violations compared to a perfect record of manual approaches