Glimpse

Continuous, Real-Time Object Recognition on Mobile Devices

Tiffany Chen

Lenin Ravindranath

Shuo Deng

Victor Bahl

Hari Balakrishnan

CSAIL

Microsoft Research

Continuous, Real-Time Recognition Apps

Driver Assistance

Augmented Reality Shopping

Face Recognition

Augmented Reality Tourist App

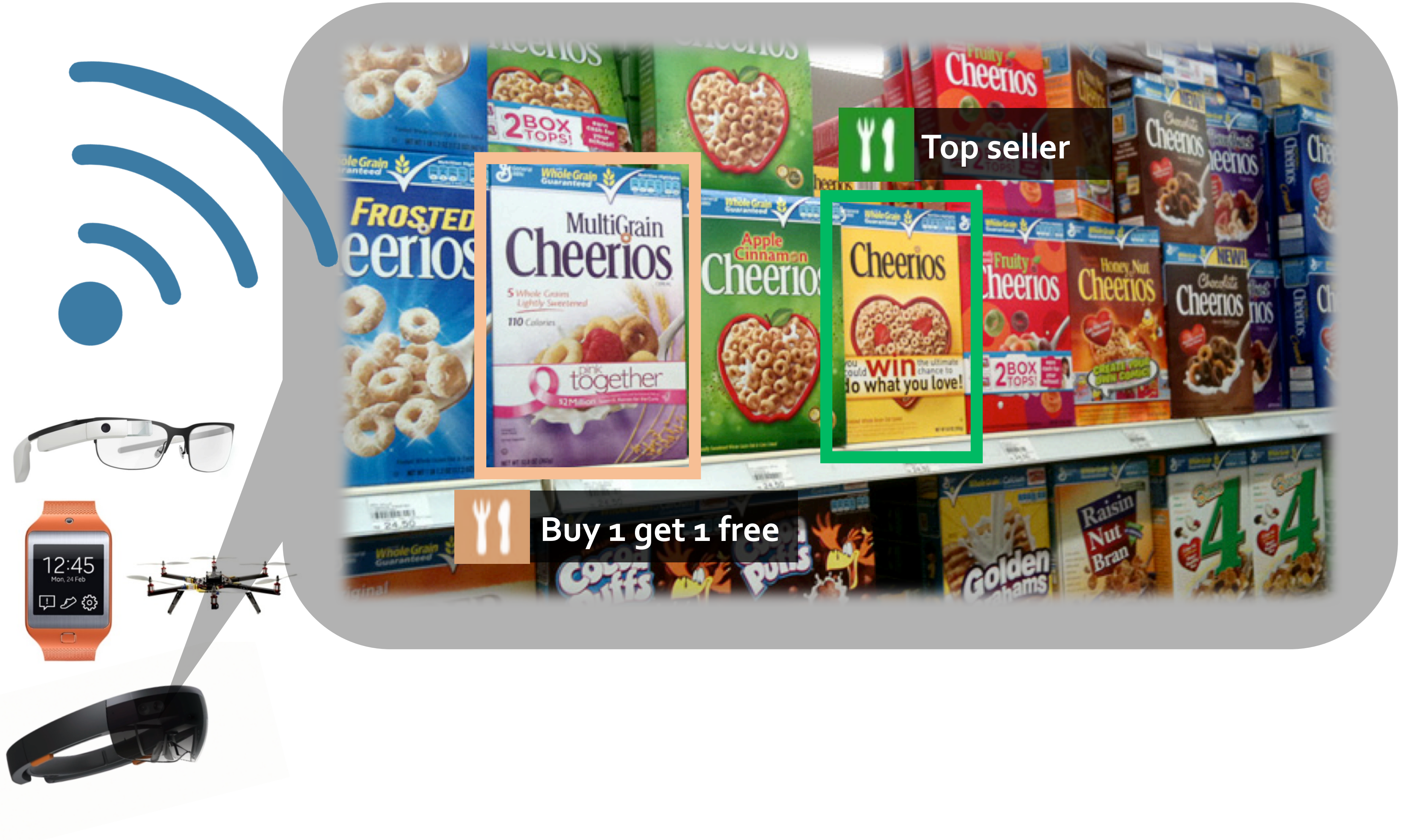
Today: Picture-Based Object Recognition

Today: Picture-Based Object Recognition

Today: Picture-Based Object Recognition

Video-Based Object Recognition

Video-Based Object Recognition

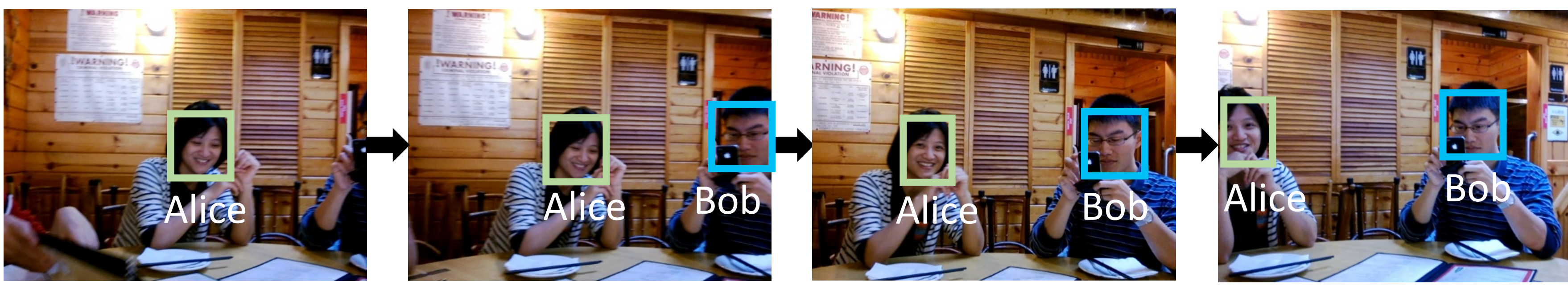


Glimpse

- Continuous, real-time object recognition on mobile devices in a video stream

Glimpse

- Continuous, real-time object recognition on mobile devices in a video stream
- Continuously ***identify*** and ***locate*** objects in each frame



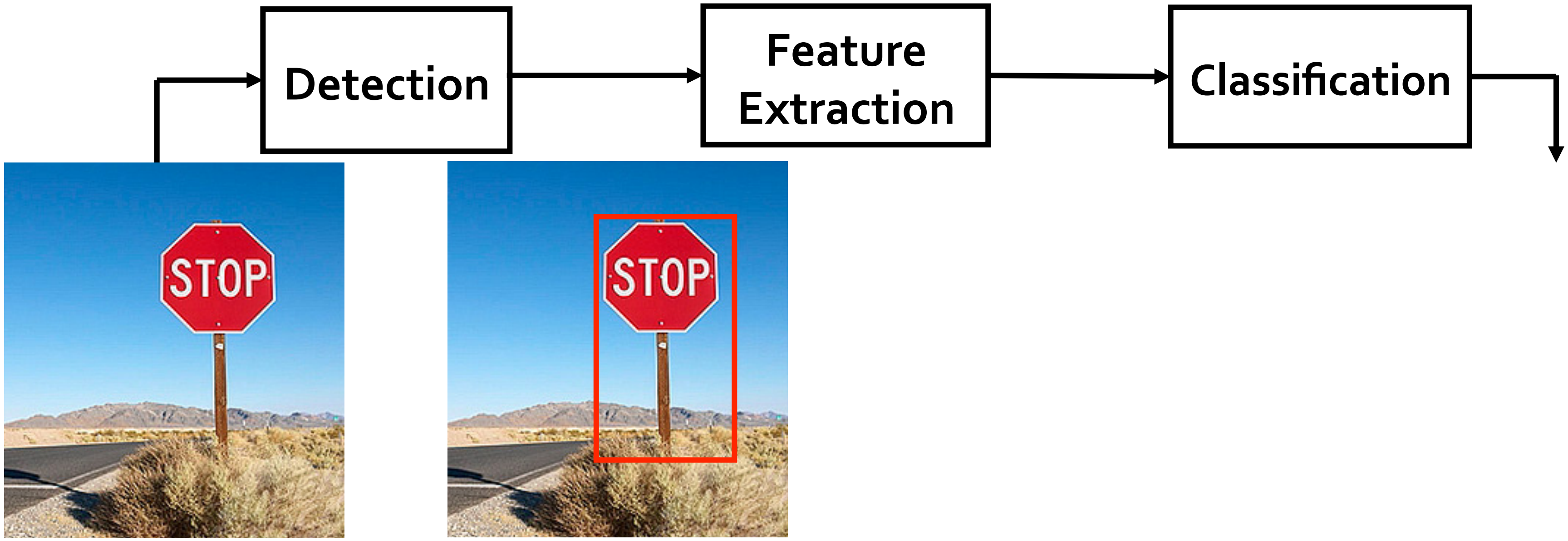
Object Recognition Pipeline

Object Recognition Pipeline

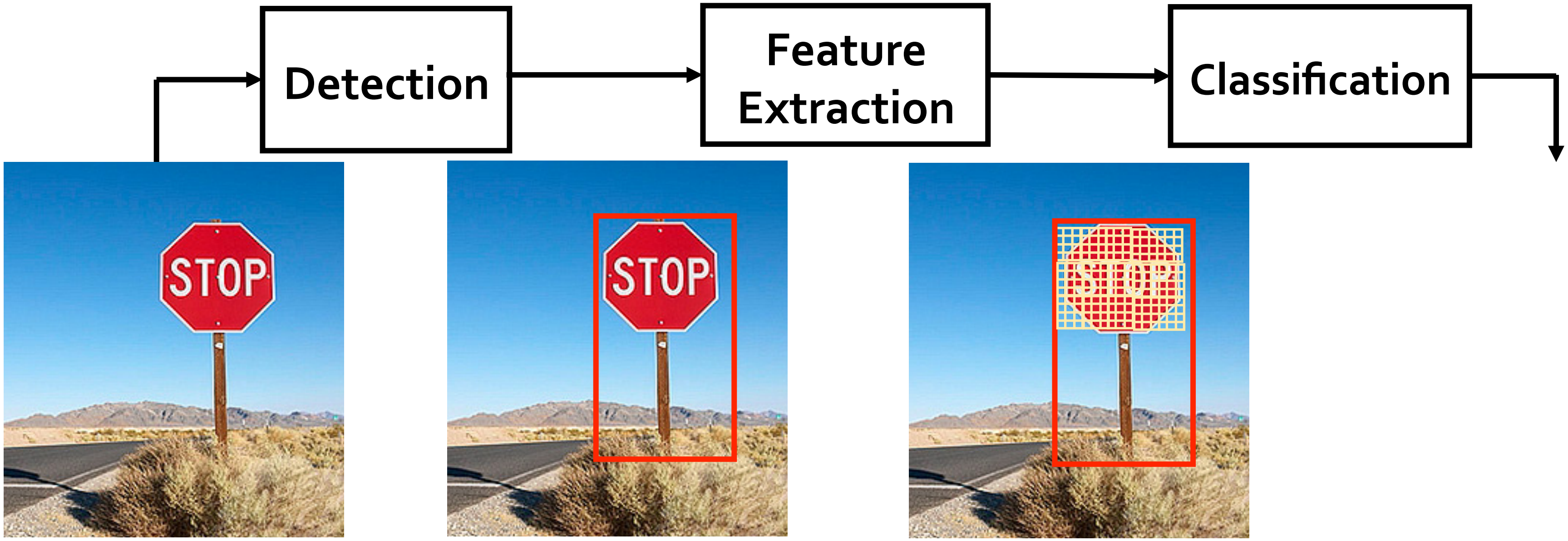
Object Recognition Pipeline

Detection → Feature Extraction → Classification

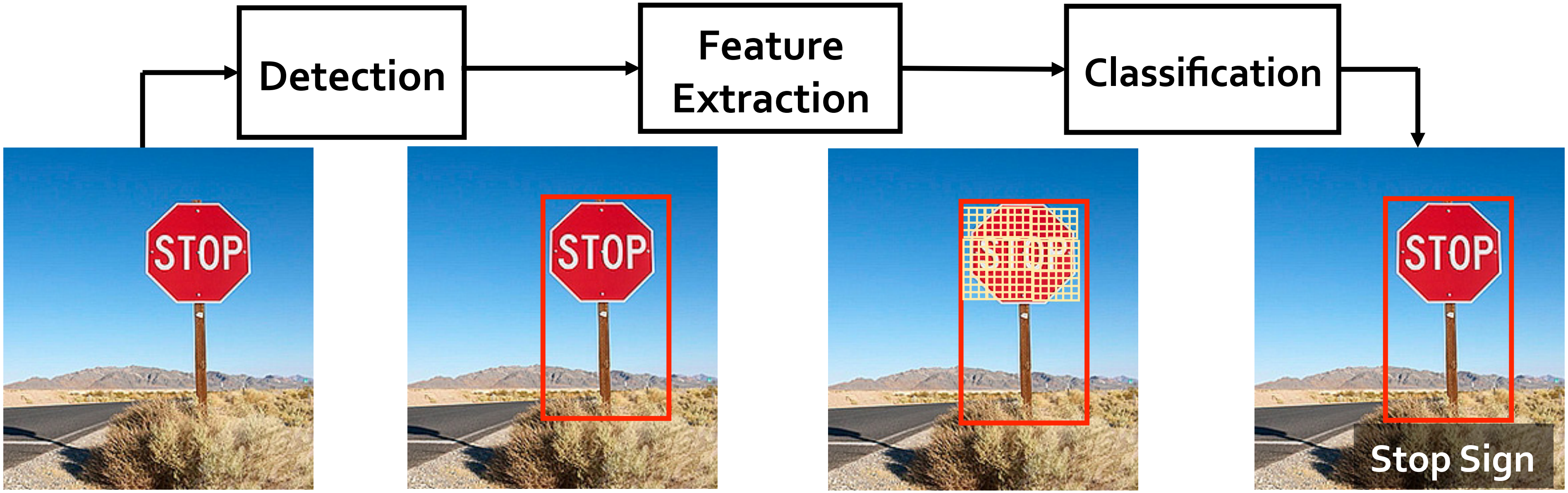
Object Recognition Pipeline



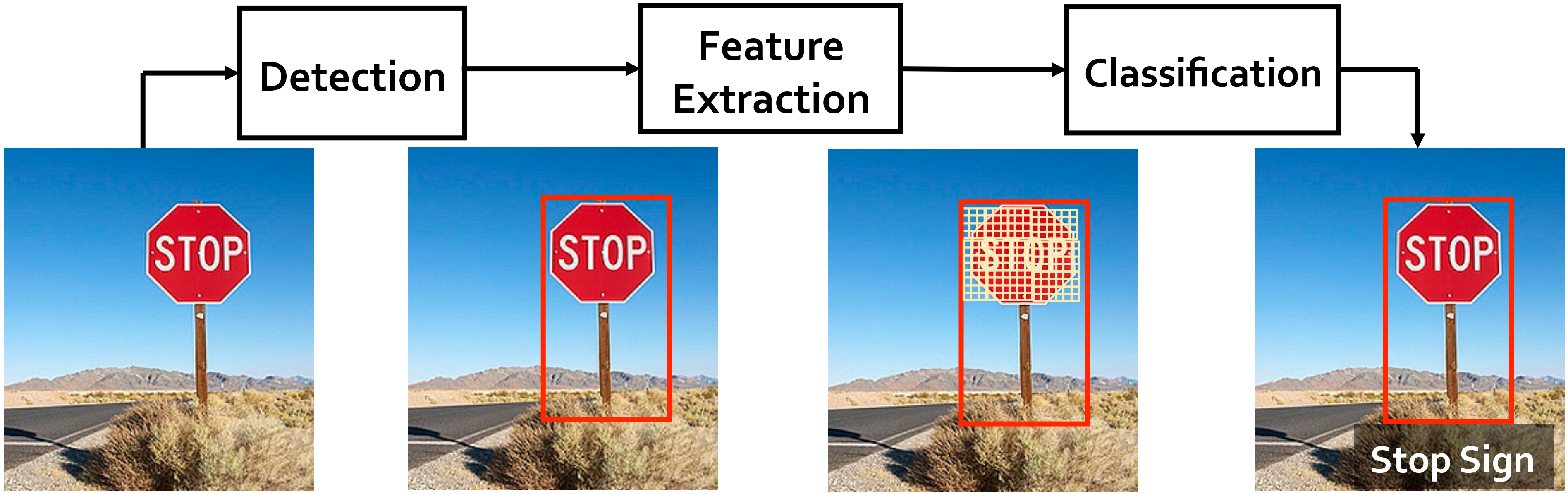
Object Recognition Pipeline



Object Recognition Pipeline

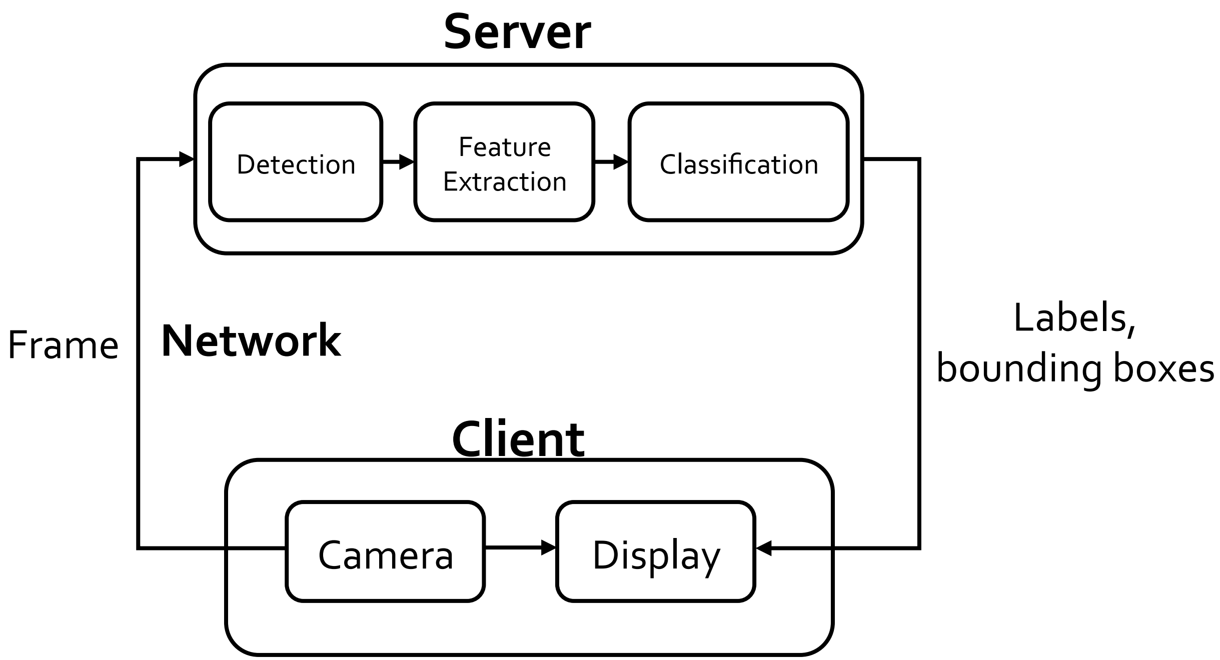


Object Recognition Pipeline

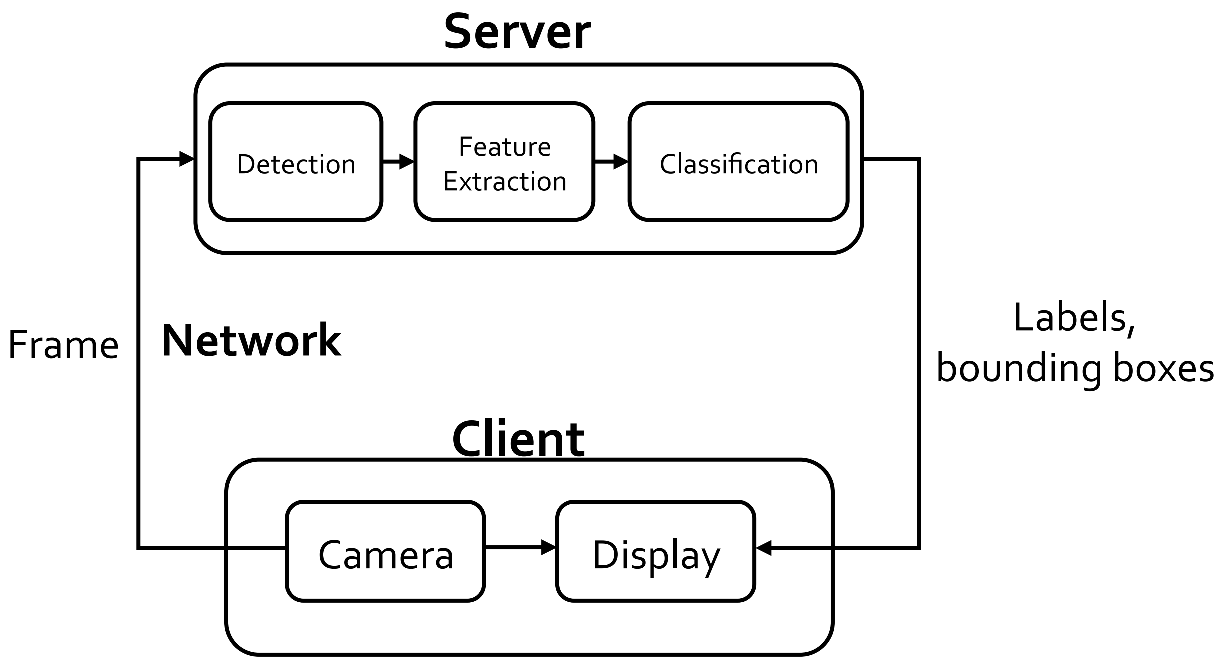


- Computationally expensive and memory-intensive
  - Server is 700x faster than Google Glass
  - Scalability
- We need to offload the recognition pipeline to servers

Client-Server Architecture



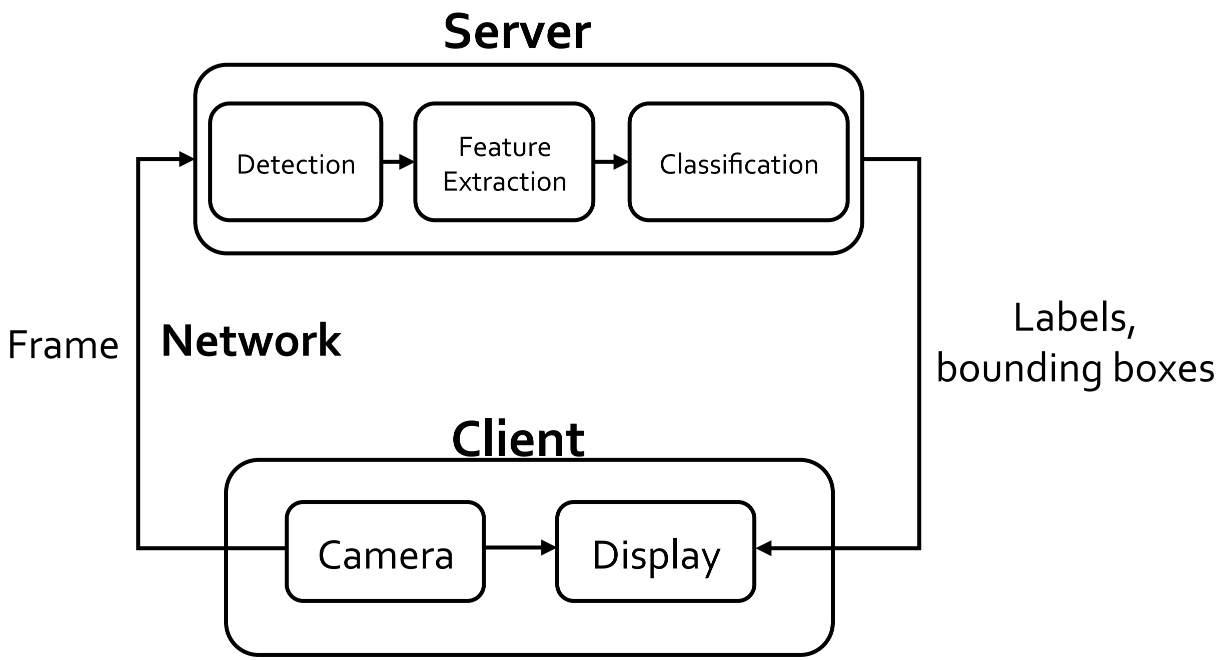
Client-Server Architecture



Challenges

1. **End-to-end latency** lowers object recognition accuracy

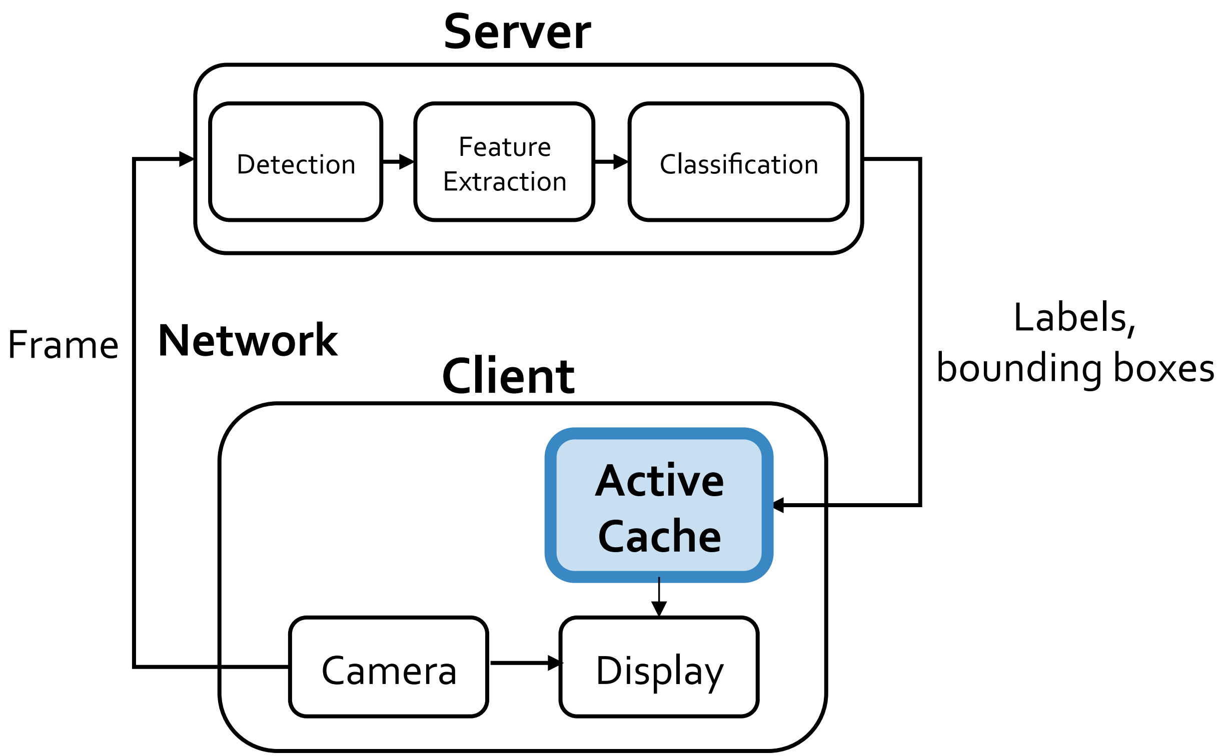
Client-Server Architecture



Challenges

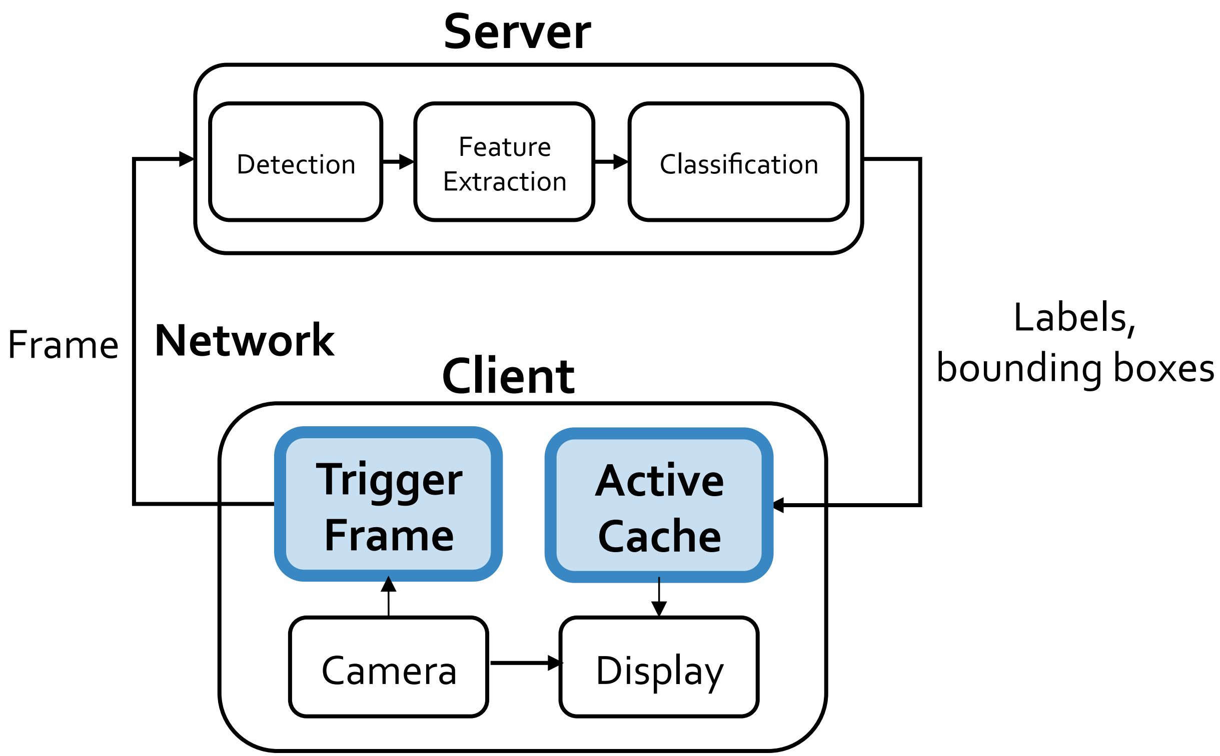
1. **End-to-end latency** lowers object recognition accuracy
2. **Bandwidth and battery** efficiency

Glimpse Architecture



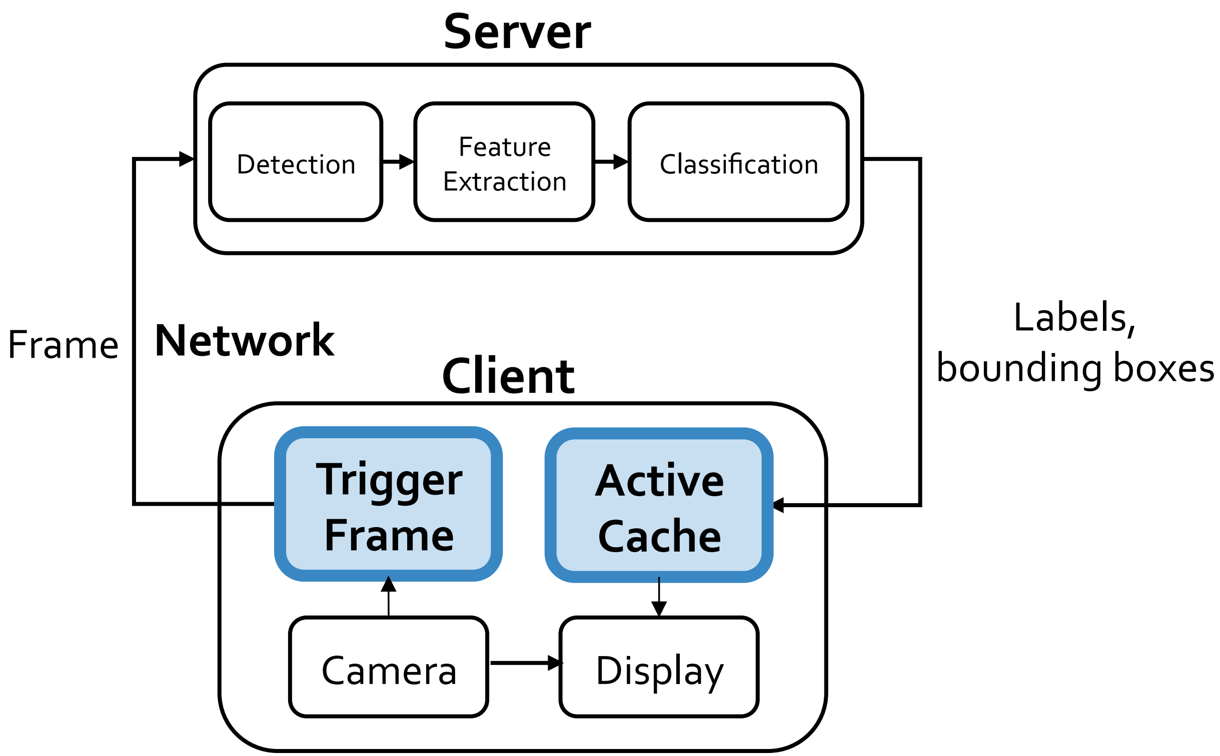
1. **Active Cache** combats e2e latency and regains accuracy

Glimpse Architecture



1. **Active Cache** combats e2e latency and regains accuracy
2. **Trigger Frame** reduces bandwidth usage

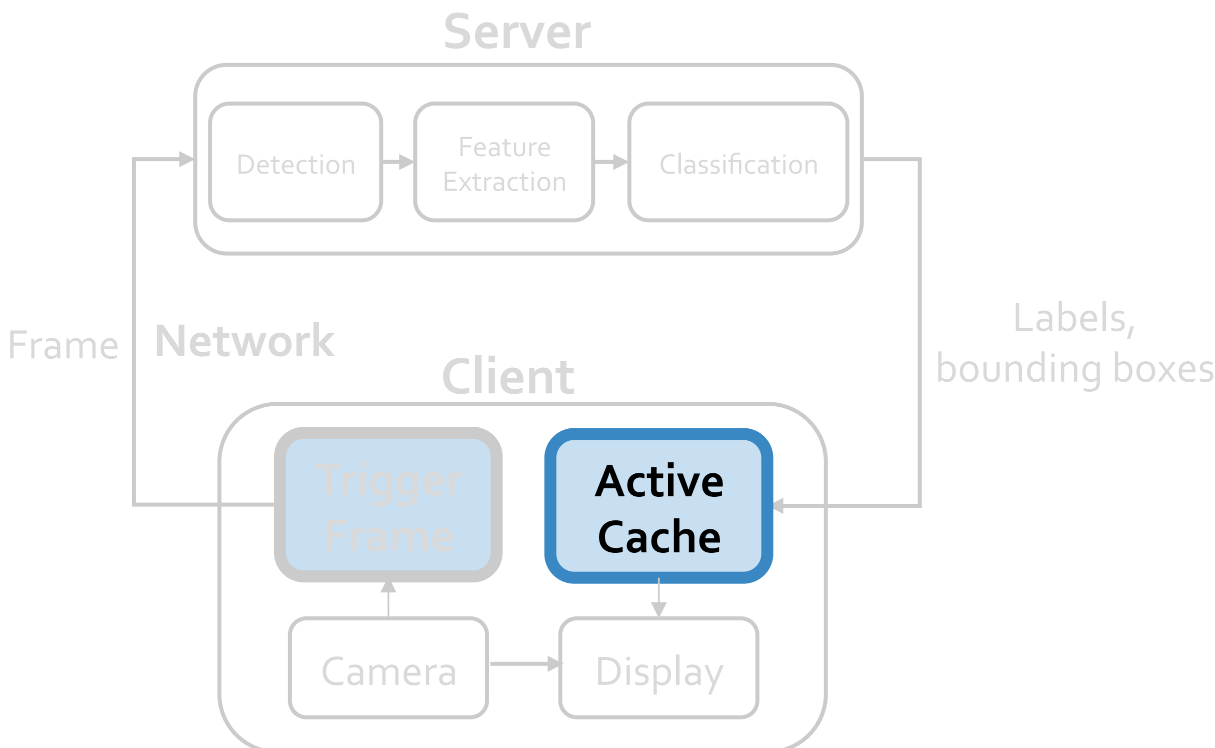
Glimpse Architecture



1. **Active Cache** combats e2e latency and regains accuracy
2. **Trigger Frame** reduces bandwidth usage

Wishbone [NSDI'09], Maui [MobiSys'10], Odessa [MobiSys'11], Gabriel [MobiSys'13]

Glimpse Architecture



1. **Active Cache** combats e2e latency and regains accuracy

End-to-End Latency Lowers Accuracy

Expected

In reality…

End-to-End Latency Lowers Accuracy

Is it possible to combat latency and regain accuracy?

Relocate Moving Object with Tracking

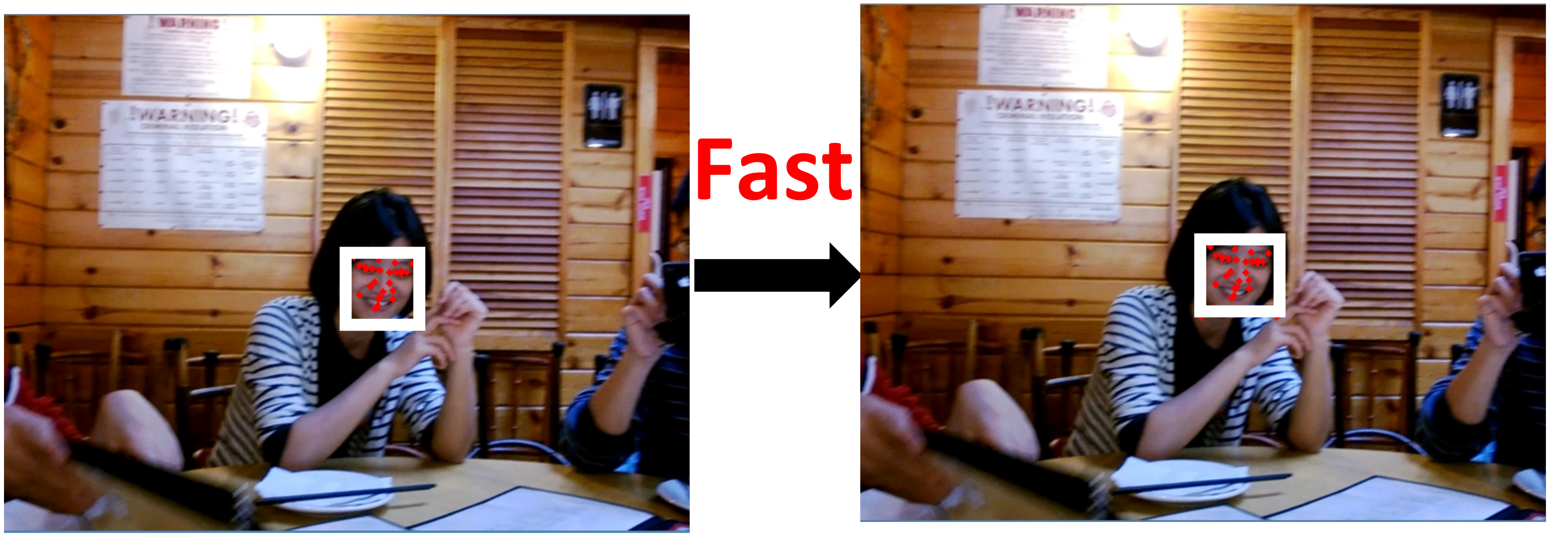
- Object tracking on the client to re-locate the object

Frame 0

Frame 12 (delay = 360 ms)

Relocate Moving Object with Tracking

- Object tracking on the client to re-locate the object



Frame 0

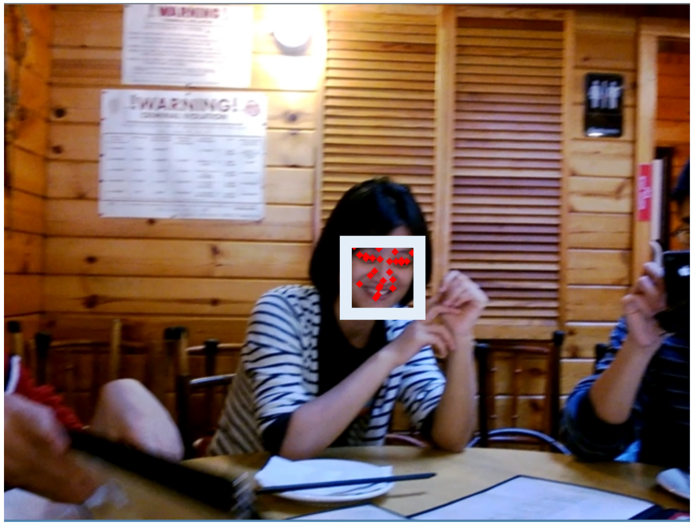
Frame 12 (delay = 360 ms)

Relocate Moving Object with Tracking

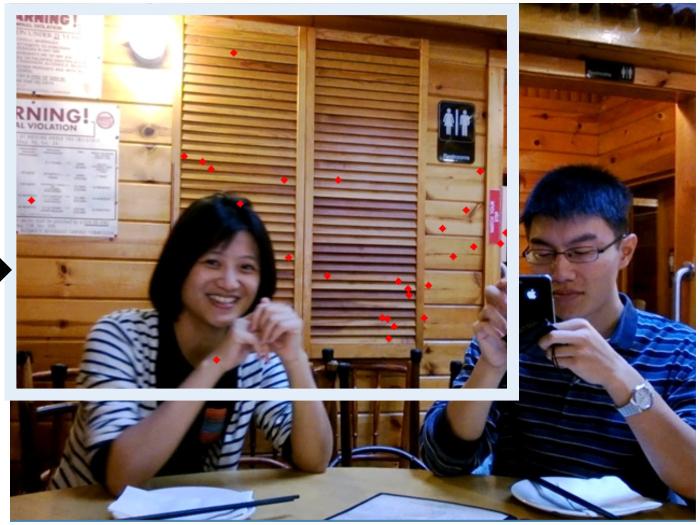
- Object tracking on the client to re-locate the object
- Fails to work when object displacement is large

Relocate Moving Object with Tracking

- Object tracking on the client to re-locate the object
- Fails to work when object displacement is large



Frame 0



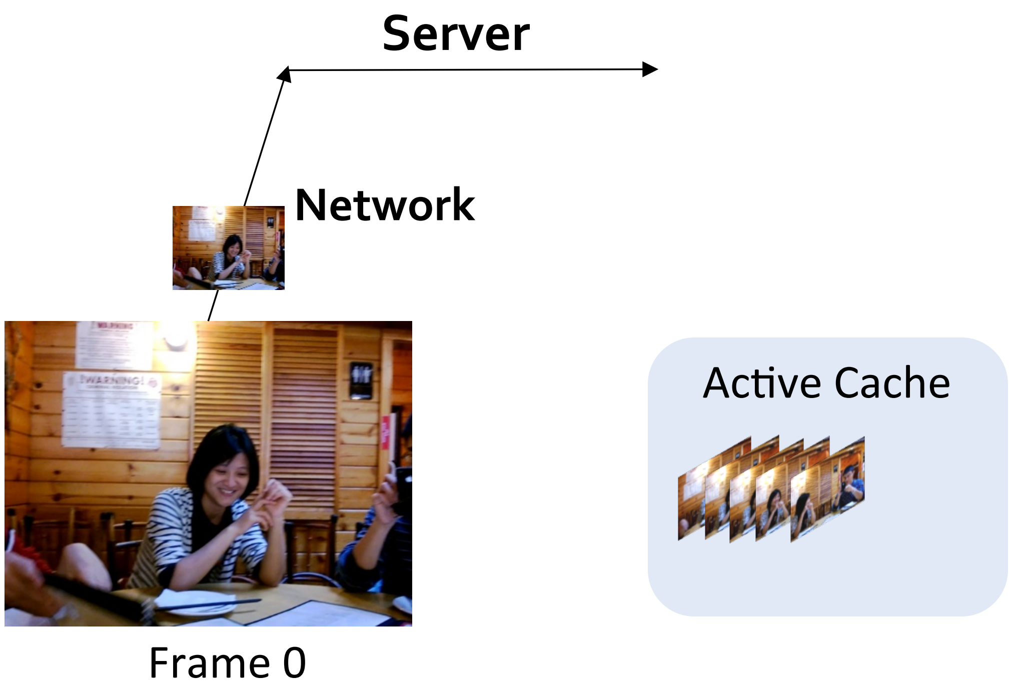
Frame 30 (delay= 1 sec)

Regain Accuracy with *Active Cache*

- Cache and run tracking through the cached frames

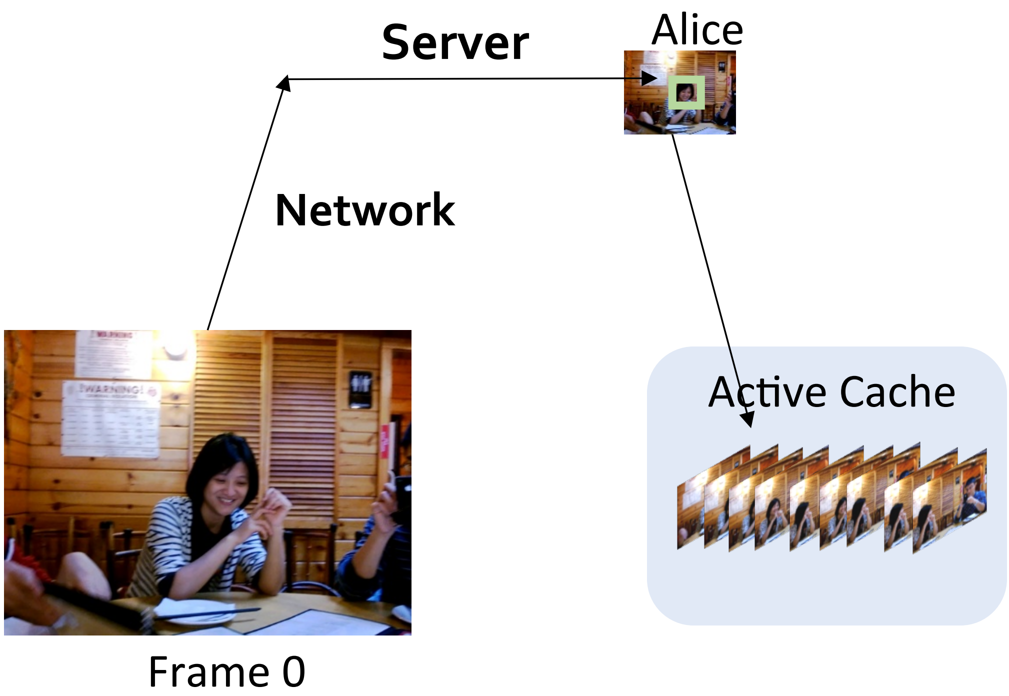
Regain Accuracy with *Active Cache*

- Cache and run tracking through the cached frames



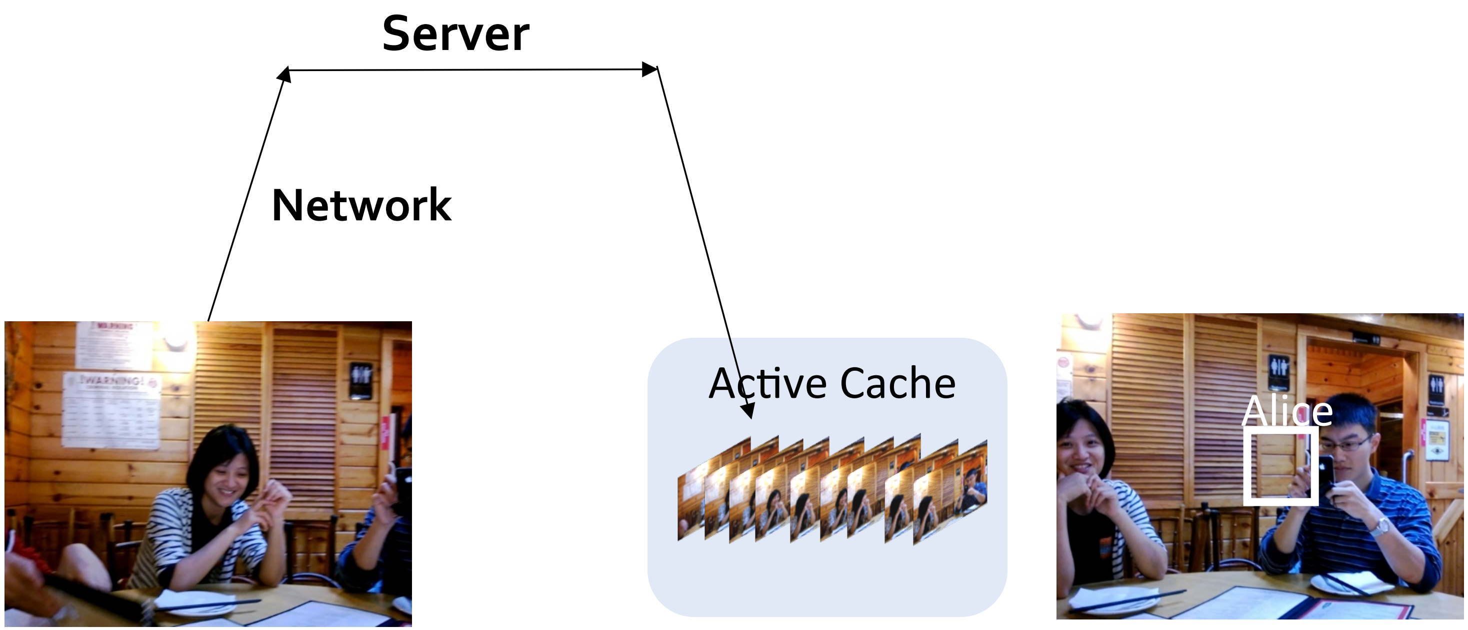
Regain Accuracy with *Active Cache*

- Cache and run tracking through the cached frames



Regain Accuracy with *Active Cache*

- Cache and run tracking through the cached frames



Frame 0

Frame 30

Run tracking from Frame 0 to Frame 30

Regain Accuracy with *Active Cache*

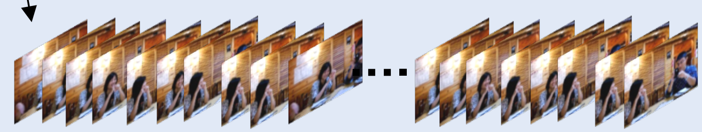
- Cache and run tracking through the cached frames

Server

Network

Tracking through all cached frames takes too long!

Active Cache



Regain Accuracy with *Active Cache*

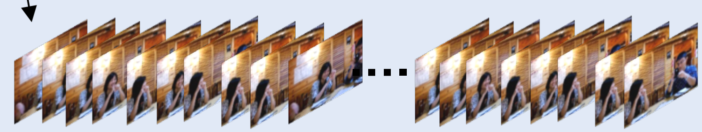
- Cache and run tracking through the cached frames

Server

Network

Tracking through all cached frames takes too long!

Active Cache



Adaptive Frame Selection

Given ***n_cached*** frames, select ***s_selected*** frames so that we can catch up without sacrificing tracking performance

Adaptive Frame Selection

Given ***n_cached*** frames, select ***s_selected*** frames so that we can catch up without sacrificing tracking performance

1. How many frames to select?
2. Which frames to select?

Adaptive Frame Selection

Given ***n_cached*** frames, select ***s_selected*** frames so that we can catch up without sacrificing tracking performance

1. **How many frames to select?**

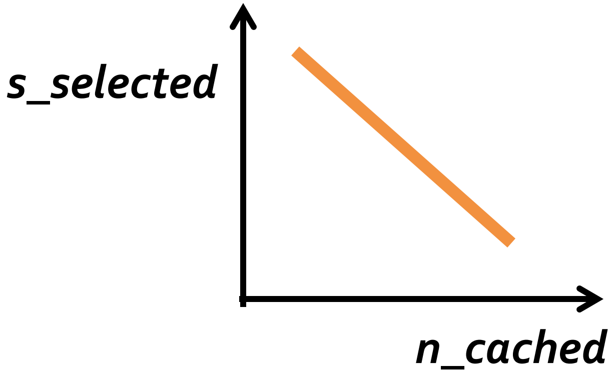
- ***s_selected:*** active cache processing time vs. tracking accuracy

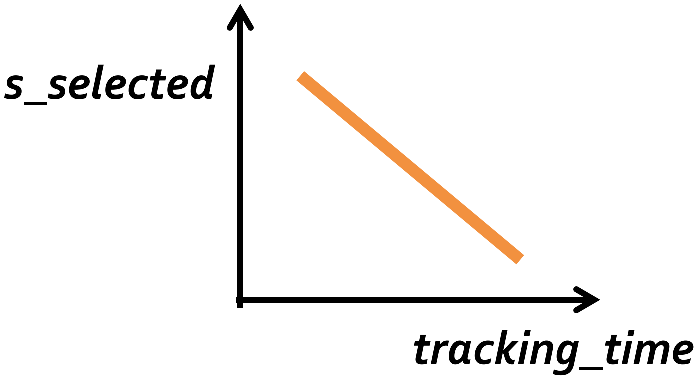
Adaptive Frame Selection

Given ***n_cached*** frames, select ***s_selected*** frames so that we can catch up without sacrificing tracking performance

1. **How many frames to select?**

- ***s_selected:*** active cache processing time vs. tracking accuracy
- ***s_selected*** depends on
  a. The end-to-end delay -- ***n_cached***
  b. The exec time of tracking on the client-- ***tracking_time***





Adaptive Frame Selection

Given ***n_cached*** frames, select ***s_selected*** frames so that we can catch up without sacrificing tracking performance

1. **How many frames to select?**

- ***s_selected:*** active cache processing time vs. tracking accuracy
- ***s_selected*** depends on
  a. The end-to-end delay -- ***n_cached***
  b. The exec time of tracking on the client-- ***tracking_time***
- Simulate *n_cached*, *tracking_time*, and *s_selected,* and pick the *s_selected* that maximizes the accuracy

Adaptive Frame Selection

Given ***n_cached*** frames, select ***s_selected*** frames so that we can catch up without sacrificing tracking performance

2. **Given *s_selected*, which frames to select?**

- Temporal redundancy between frames

Adaptive Frame Selection

Given ***n_cached*** frames, select ***s_selected*** frames so that we can catch up without sacrificing tracking performance

2. Given *s_selected*, which frames to select?

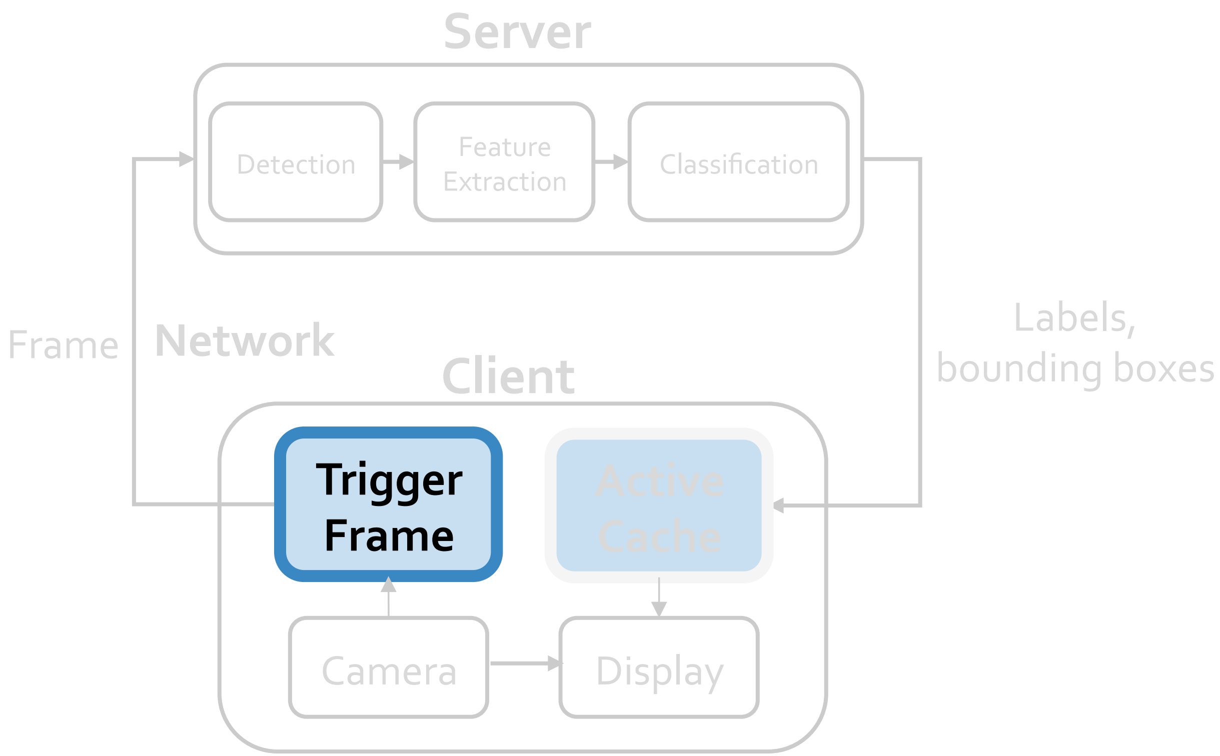
- Temporal redundancy between frames
- Use *frame differencing* to quantify movement and select frames to capture as much movement as possible

Active Cache Achieves Higher Accuracy

Before Active Cache | **After Active Cache**

- Active Cache can be applied to any objects
- Active Cache can be used to hide any end-to-end delay

Glimpse Architecture



1. **Active Cache** combats e2e latency and regains accuracy
2. **Trigger Frame** reduces bandwidth usage

Reduce Bandwidth Usage with Trigger Frames

- Strategically send certain trigger frames to the server

Reduce Bandwidth Usage with Trigger Frames

- Strategically send certain trigger frames to the server

1. Measuring scene changes

Reduce Bandwidth Usage with Trigger Frames

- Strategically send certain trigger frames to the server

1. Measuring scene changes
2. Detecting tracking failure

- The standard deviation of distance of all tracked points between two frames

Reduce Bandwidth Usage with Trigger Frames

- Strategically send certain trigger frames to the server
  1. Measuring scene changes
  2. Detecting tracking failure
- Limiting the number of frames in-flight

Evaluation

- **Object recognition pipelines**
  1. Face recognition
  2. Road sign recognition

Evaluation

- **Object recognition pipelines**
  1. Face recognition
  2. Road sign recognition
- **Datasets**

  1. Face Dataset:
  - 26 videos recorded with a smartphone
  - 30 minutes, 54K frames, and 36K faces
  - Scenarios: shopping with friends and waiting at a subway station

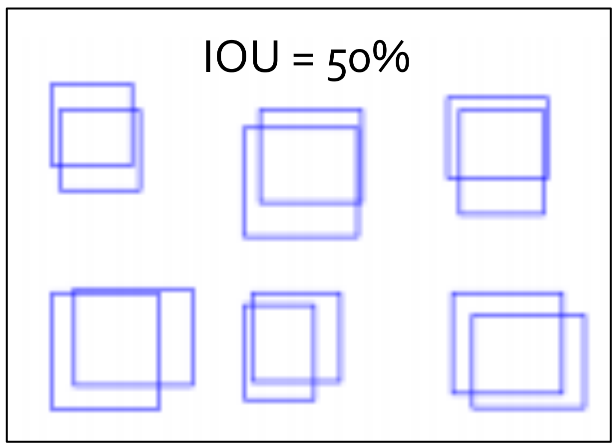
  2. Road Sign Dataset:
  - 4 walking videos recorded using Google Glass from YouTube
  - 35 minutes, 63K frames, and 5K road signs

Evaluation

- **Evaluation Metrics**
  - Intersection over union (IOU) to measure localization accuracy

$$IOU_i = \frac{area\,|O_i \cap G_i|}{area\,|O_i \cup G_i|}$$

O*i:* bounding box of the detected object *i*
G*i:* bounding box of object *i*'s ground truth



  - **Correct if <u>IOU > 50%</u> and the <u>label matches</u> ground truth**

Evaluation

- **Evaluation Metrics**
  - Precision

$$\frac{\text{\# of objects correctly labeled and located}}{\text{total \# of objects detected}}$$

  - Recall

$$\frac{\text{\# of objects correctly labeled and located}}{\text{total \# of objects in the ground truth}}$$

Evaluation

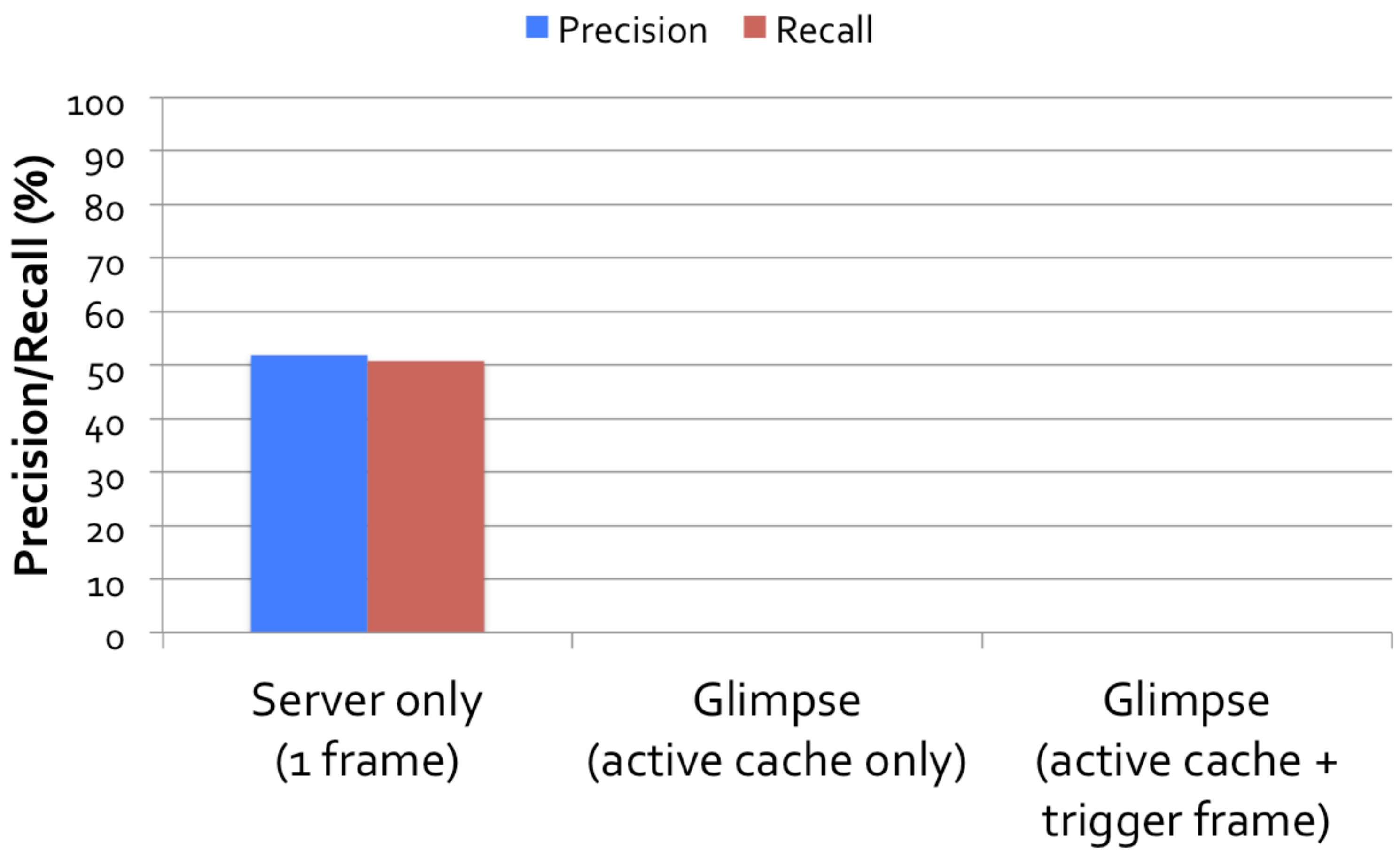
- **Network conditions**
  - Wi-Fi, Verizon’s LTE, and AT&T’s LTE network

Results Outline

1. Face recognition
2. Road sign recognition
3. Face recognition with hardware-assisted face detection

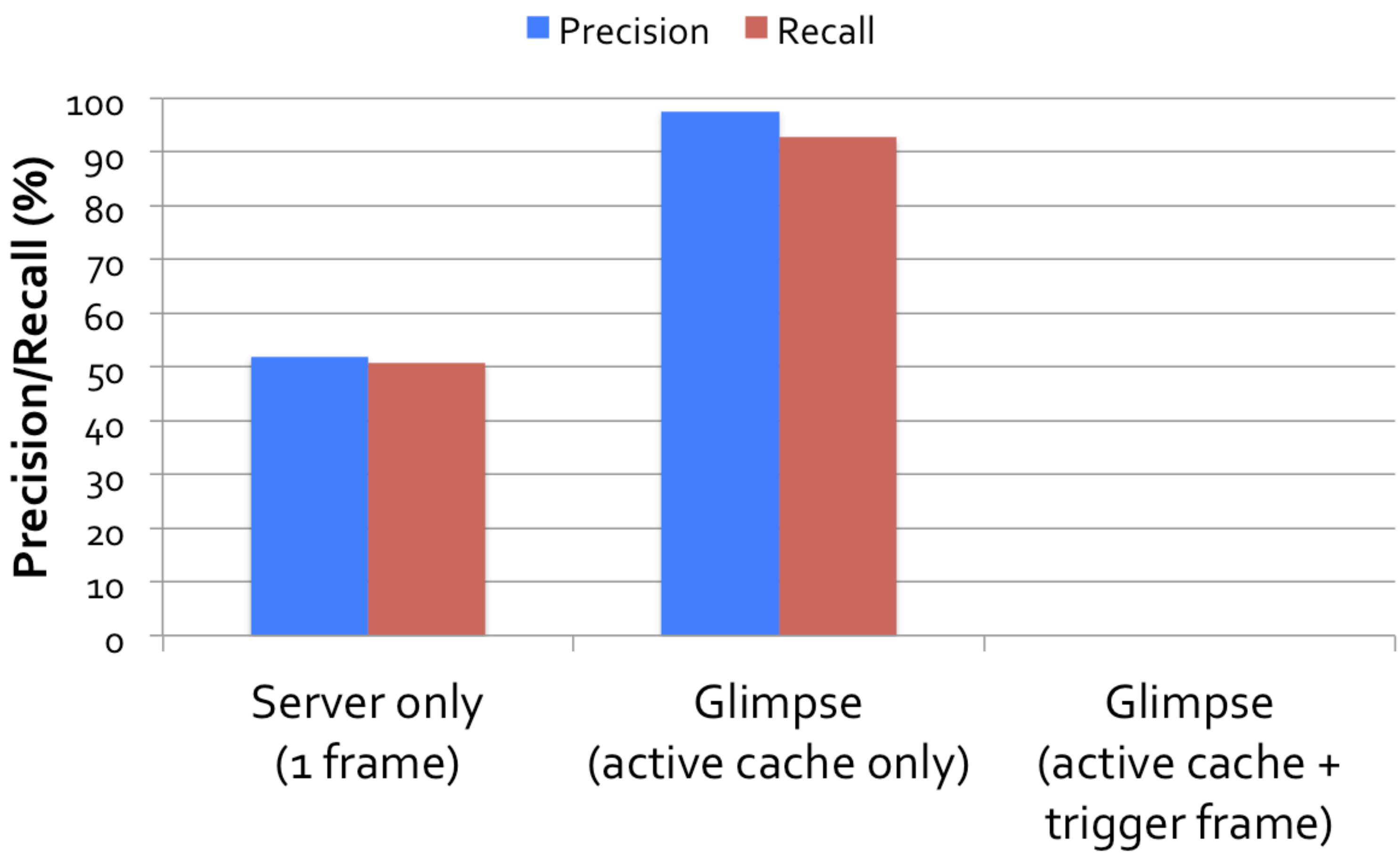
Active Cache Achieves High Accuracy

- Face dataset
- Wi-Fi (End-to-end delay: 430 ms)



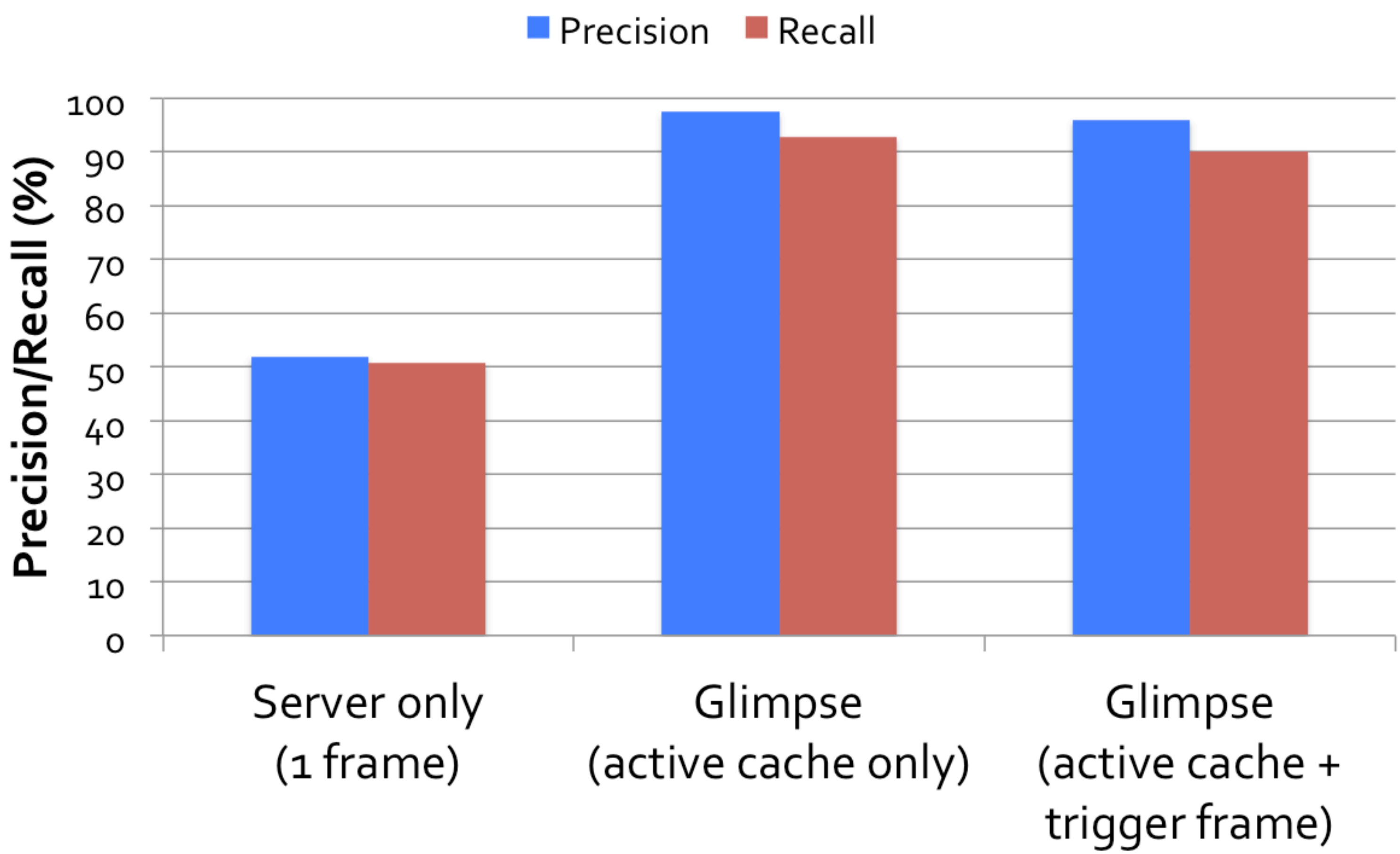
Active Cache Achieves High Accuracy

- Face dataset
- Wi-Fi (End-to-end delay: 430 ms)



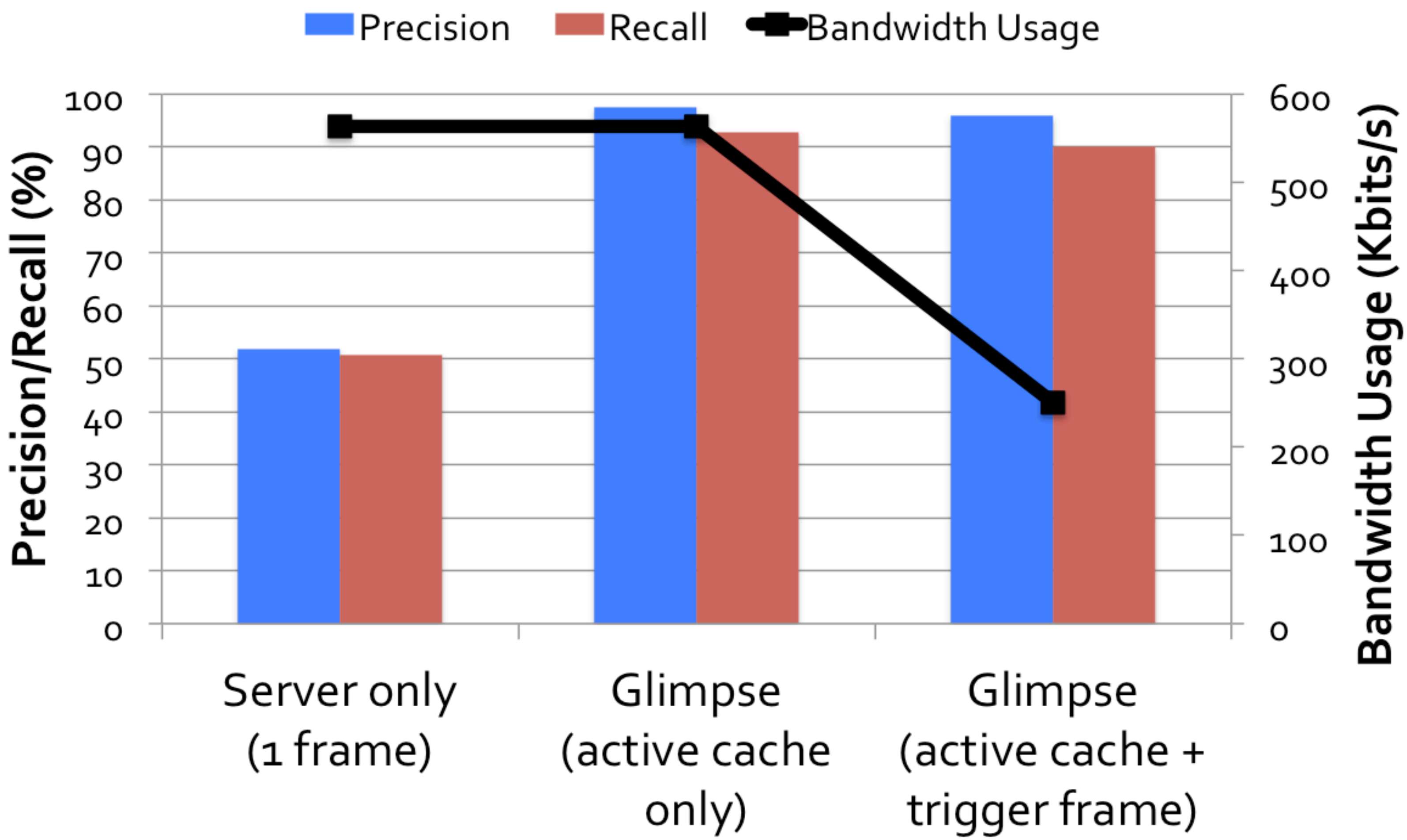
Trigger Frame Reduces Bandwidth Usage without Sacrificing Accuracy

- Face dataset
- Wi-Fi (End-to-end delay: 430 ms)



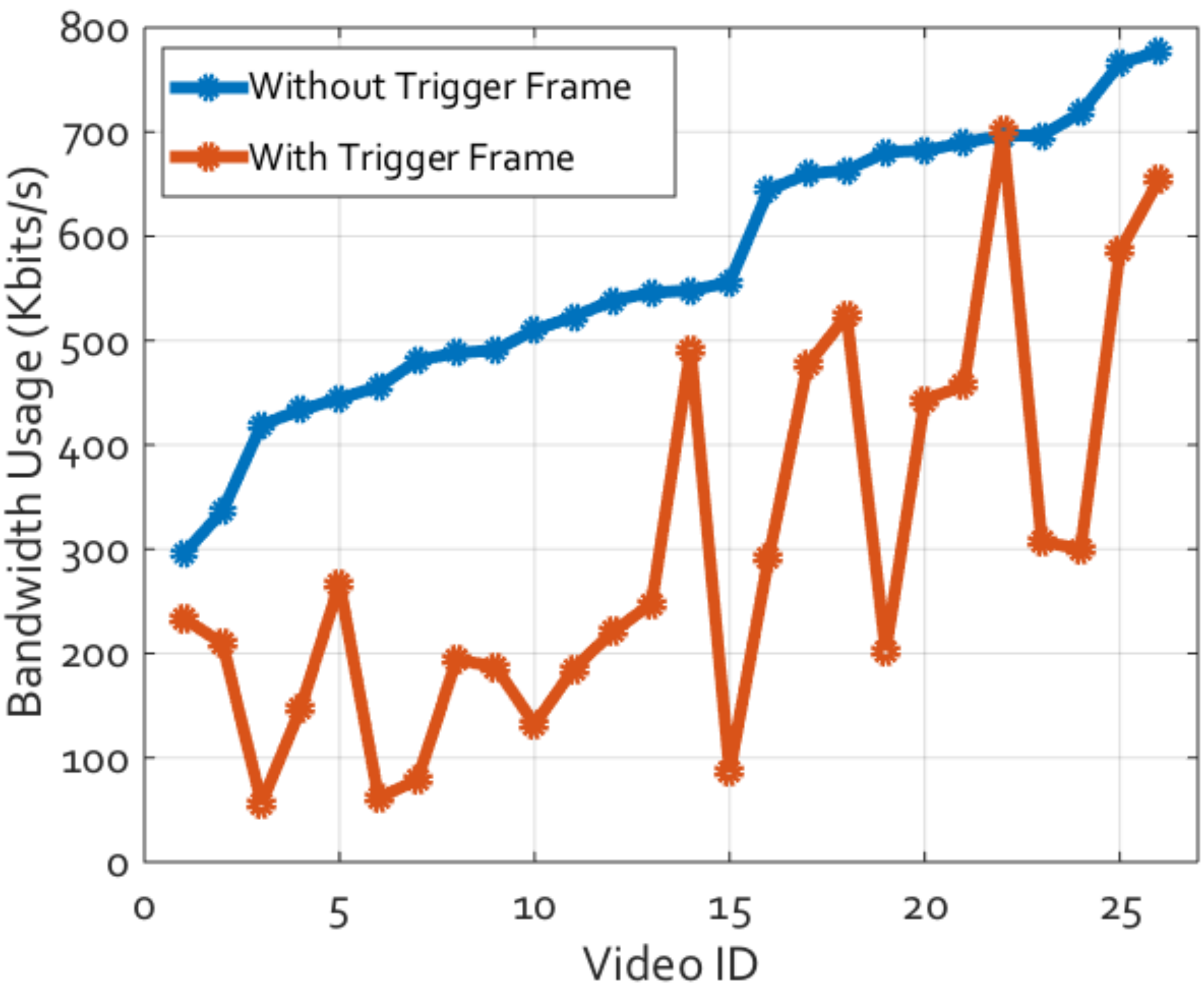
Trigger Frame Reduces Bandwidth Usage without Sacrificing Accuracy

- Face dataset
- Wi-Fi (End-to-end delay: 430 ms)



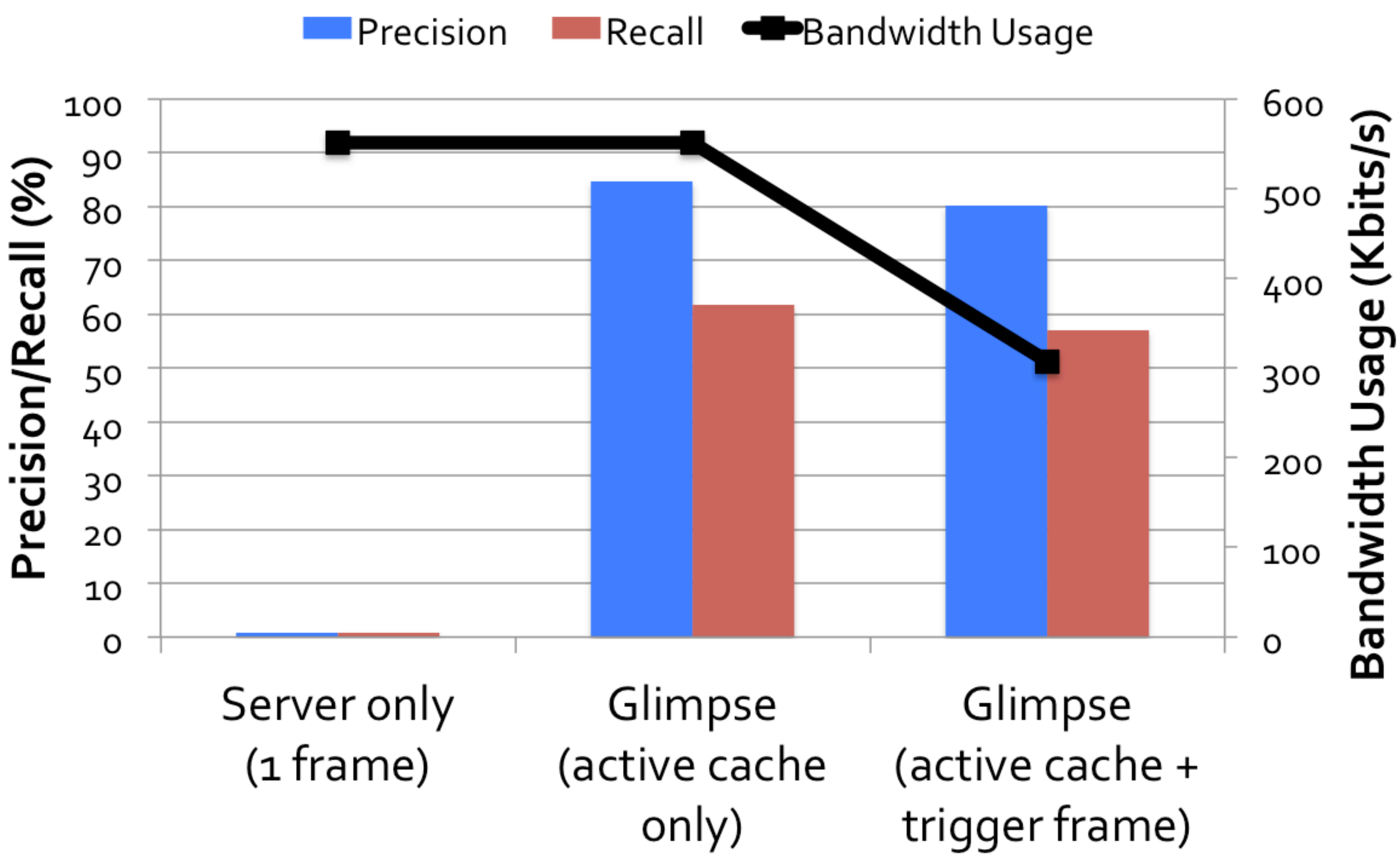
Trigger Frame Consistently Reduces Bandwidth Usage

- Face Dataset (Wi-Fi)



Glimpse Achieves Higher Accuracy and Lower Bandwidth Usage

- Road sign dataset
- Wi-Fi (End-to-end delay: 520 ms)

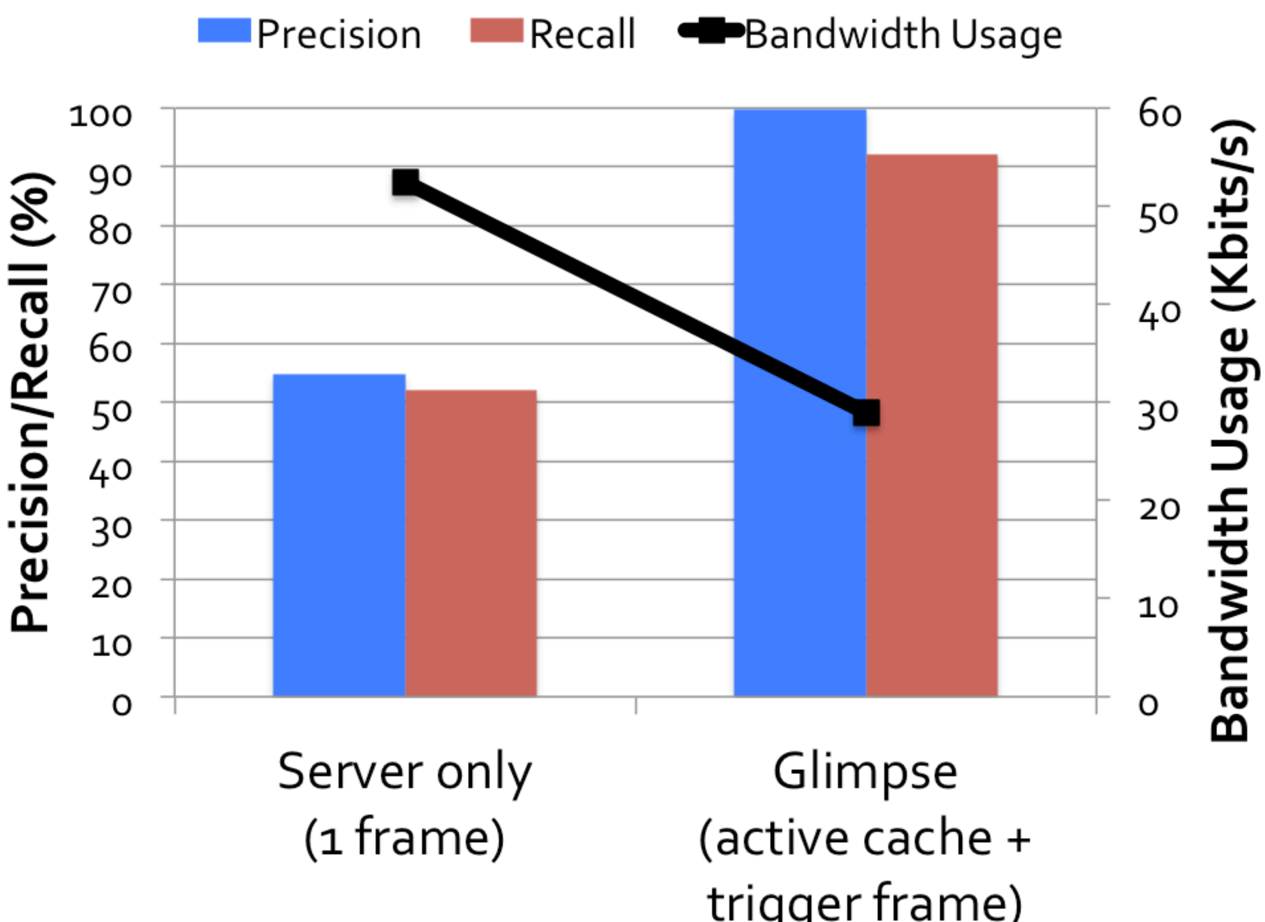


Hardware-Assisted Object Detection

- Mobile devices are now equipped with object detection hardware
- Is Glimpse still helpful?

Glimpse Improves Accuracy even with Detection Hardware on Devices

- Face dataset (Wi-Fi)
- Face detection in hardware



Glimpse

- Glimpse enables continuous, real time object recognition on mobile devices
- Glimpse achieves high recognition accuracy by maintaining an *active cache* of frames on the client
- Glimpse reduces bandwidth consumption by strategically sending only certain *trigger frames*

Active Cache and Trigger Frame are Generic

- Latency caused performance degradation and excessive resource usage are fundamental problems to object recognition
- *Active Cache* can hide any end-to-end latency
- *Trigger Frame* can reduces resource consumed