Reductо: On-Camera Filtering for Resource-Efficient Real-Time Video Analytics

Yuanqi Li, Arthi Padmanabhan, Pengzhan Zhao, Yufei Wang, Harry Xu, Ravi Netravali

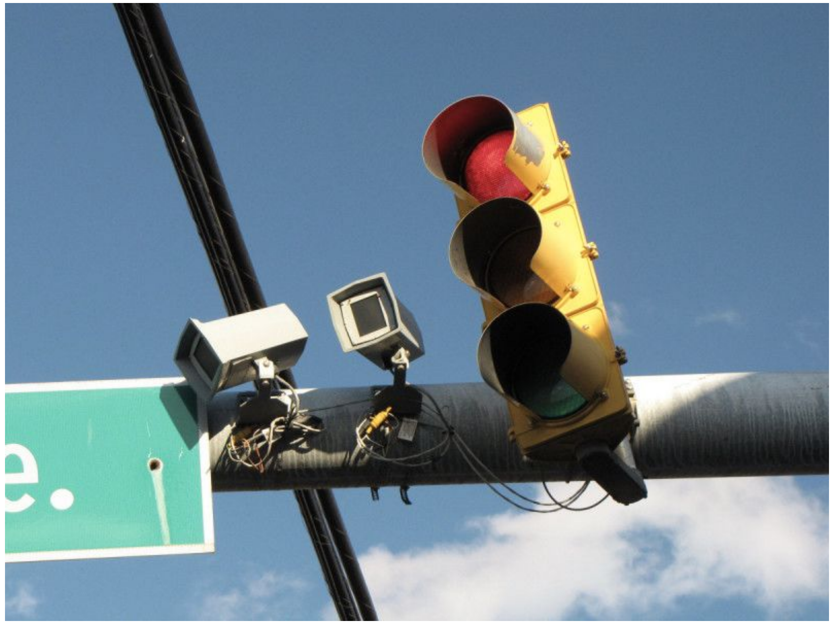
UCLA

Presented by Hongpeng Guo

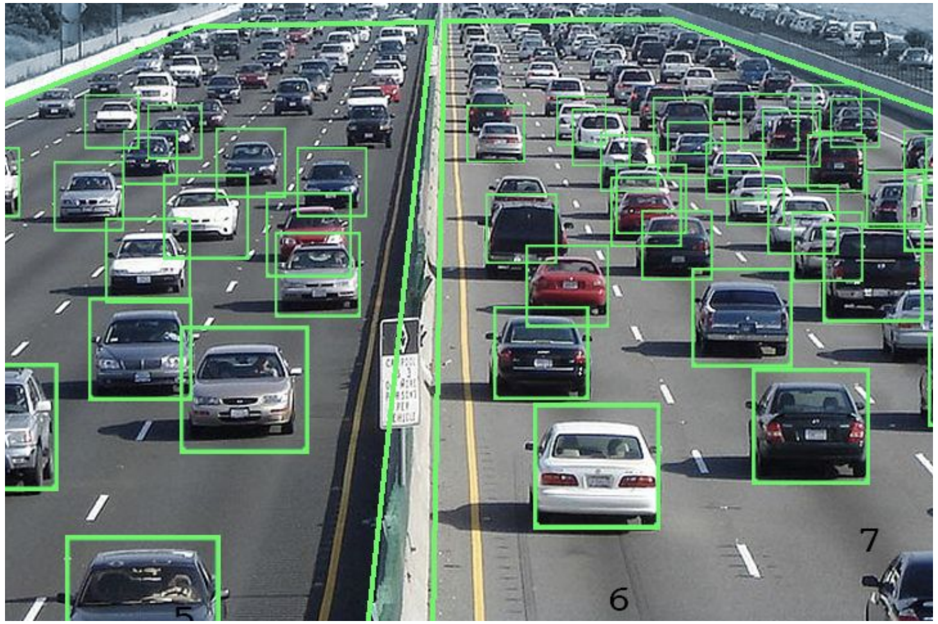
Slides are based on SIGCOMM20 presentation from Arthi Padmanabhan.

Video Analytics Trends

- More cameras and video data

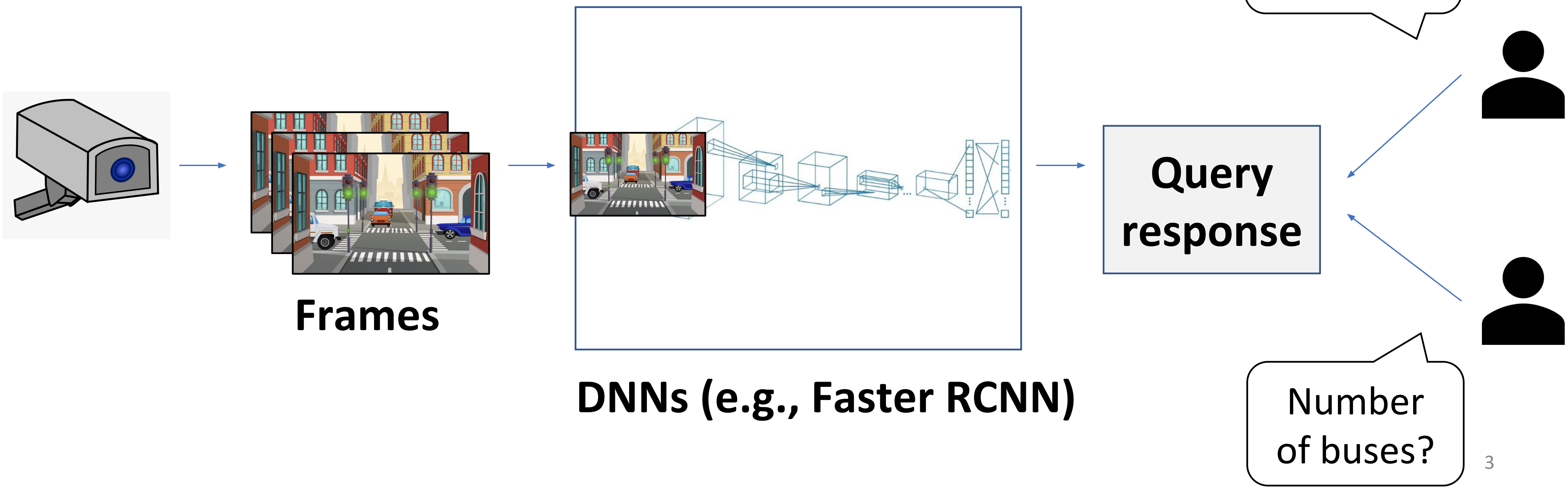


- Greater ability to extract information from video



Video Analytics Pipelines

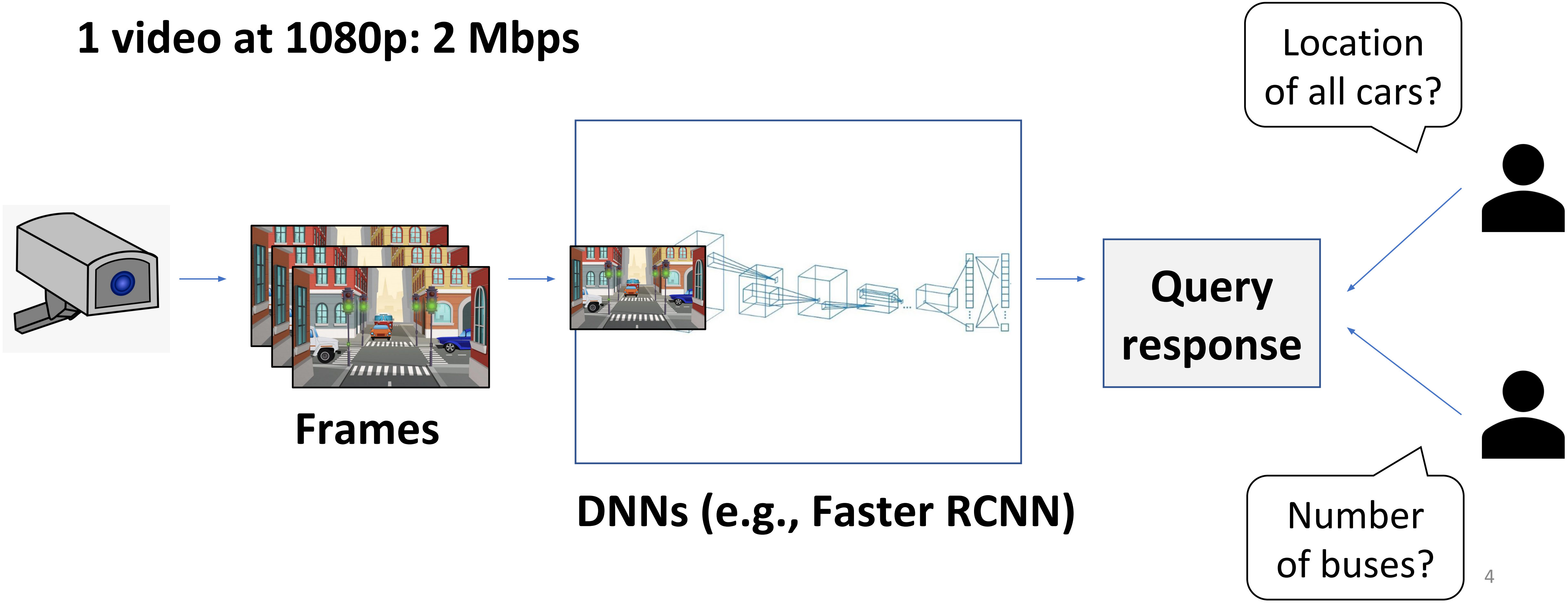
Goals:
- **Accuracy target (e.g., 90%)**
- **Real-time (e.g., 30 fps)**



Video Analytics Pipelines

Resource Intensive!

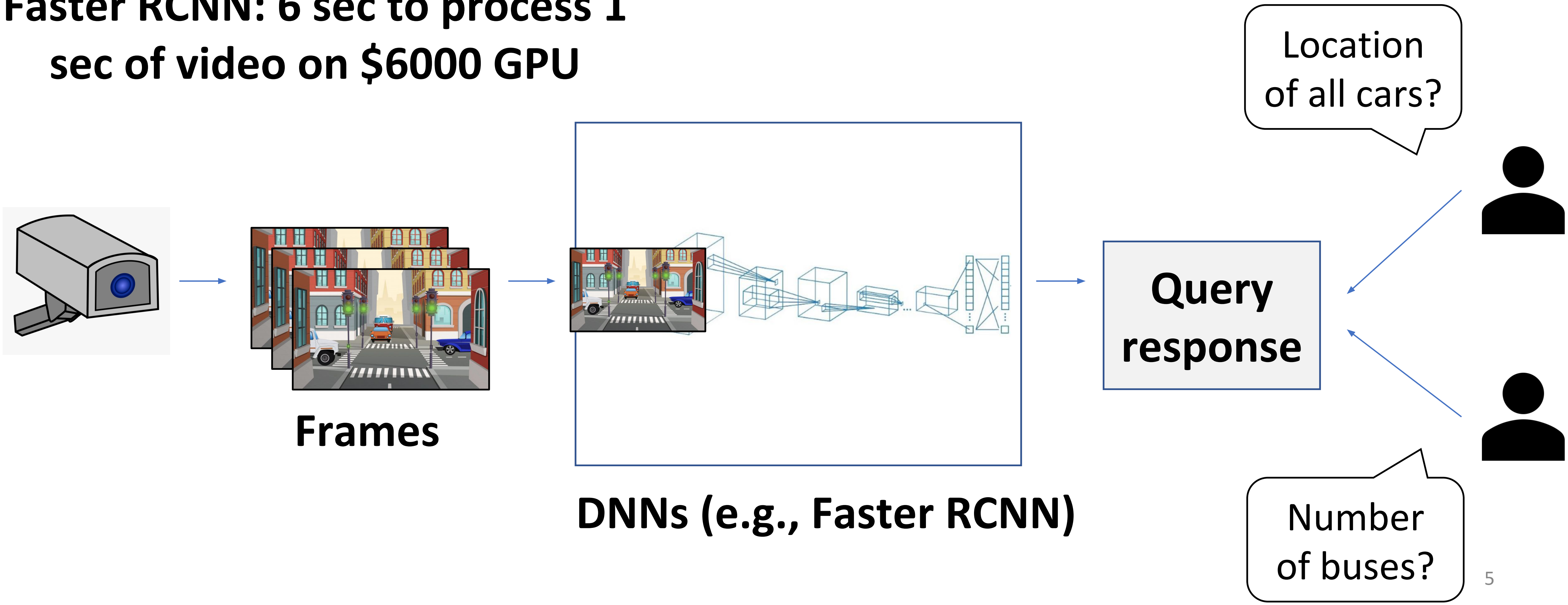
1 video at 1080p: 2 Mbps



Video Analytics Pipelines

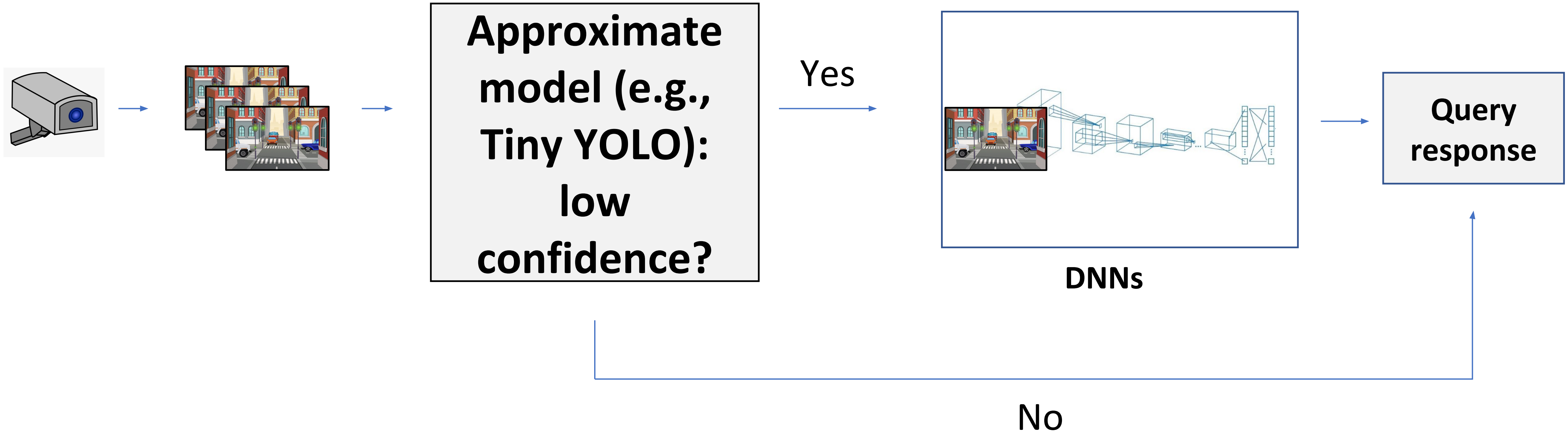
Resource Intensive!

Faster RCNN: 6 sec to process 1 sec of video on $6000 GPU



Frame Filtering

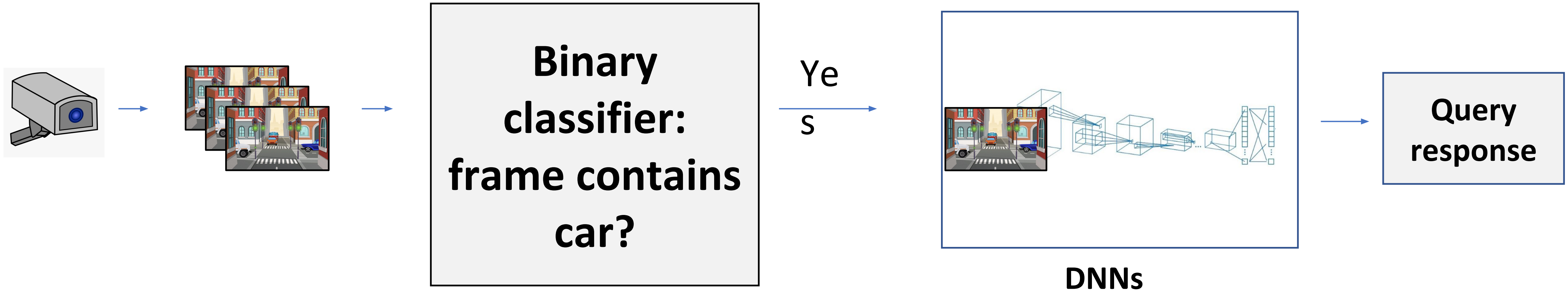
1) Approximate model
(Focus, OSDI ‘18)



Frame Filtering

1) Approximate model

2) Binary classifier (NoScope, VLDB '17)

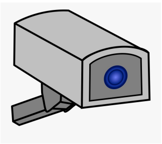


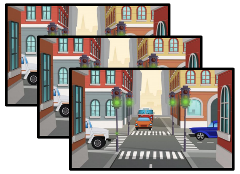
Frame Filtering

1) Approximate model

2) Binary classifier

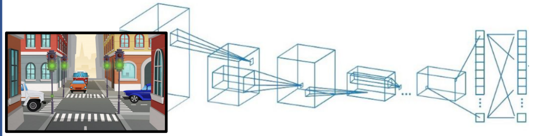
3) Pixel-level differences (Glimpse, SenSys ’15)





Pixel-level differences: frame change above threshold?

Yes



DNNs

Query response

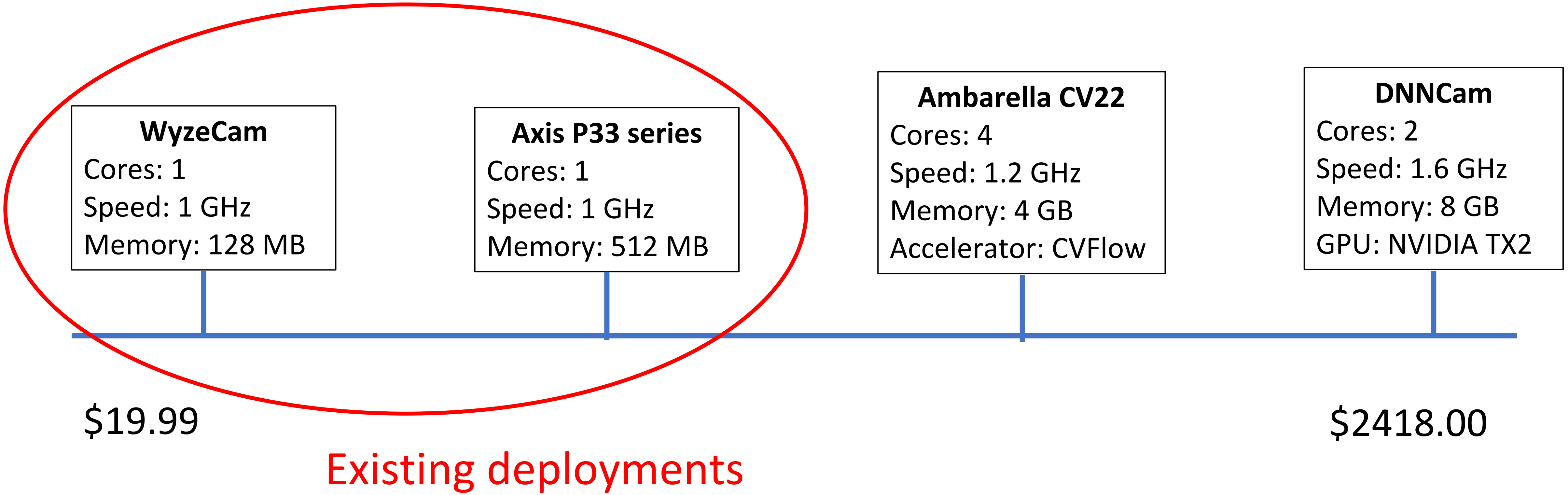
Key Question

- Filtering benefits increase closer to the video source

Can we filter frames directly on the camera itself?

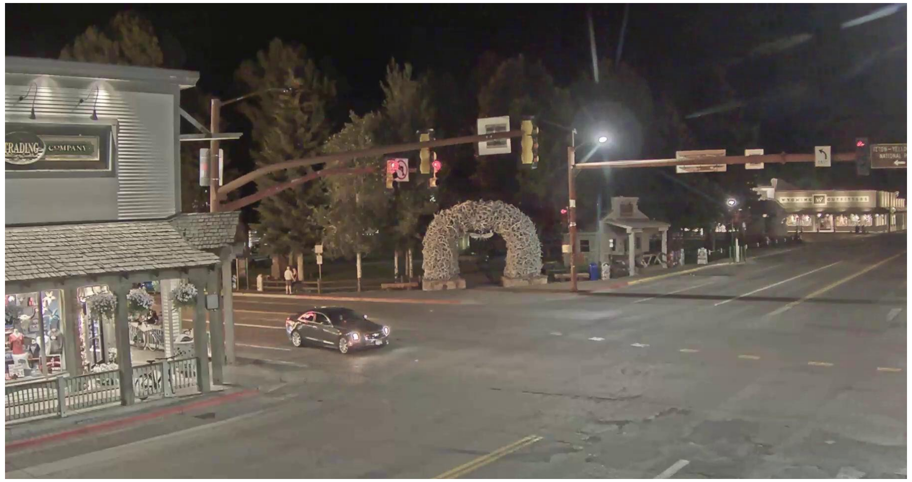
- What computational resources are available on existing cameras?
- How do existing approaches fare?

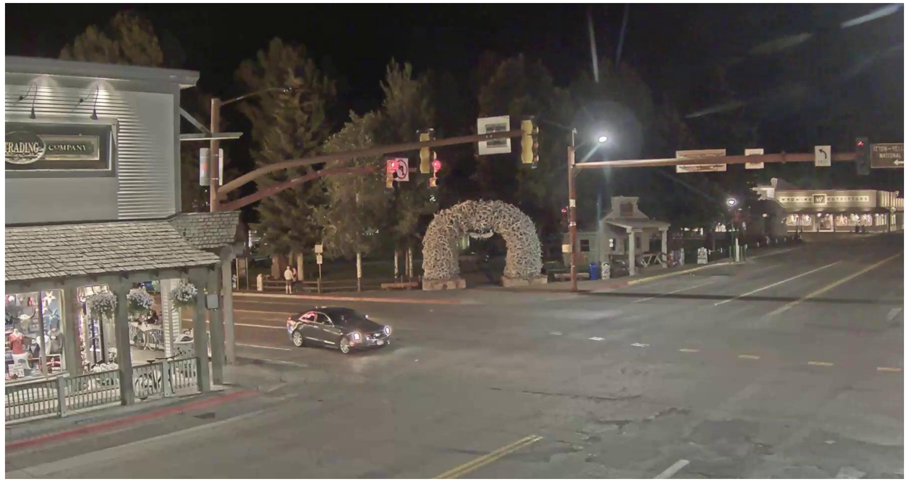
Camera Market Study



Existing Filtering Approaches

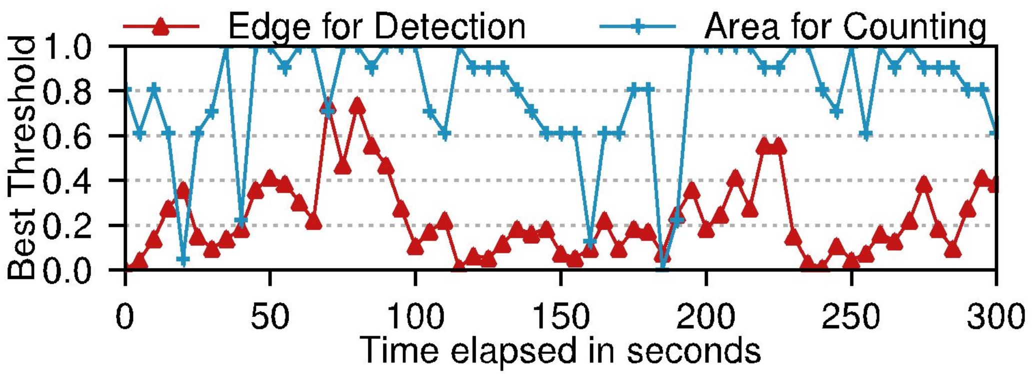
- Approximate models: too slow on camera (Tiny YOLO: 0.6 fps)
- Binary classification – misses 45% of filtering opportunities





Using Frame Differencing Effectively

- Dynamic threshold to deal with rapid changes



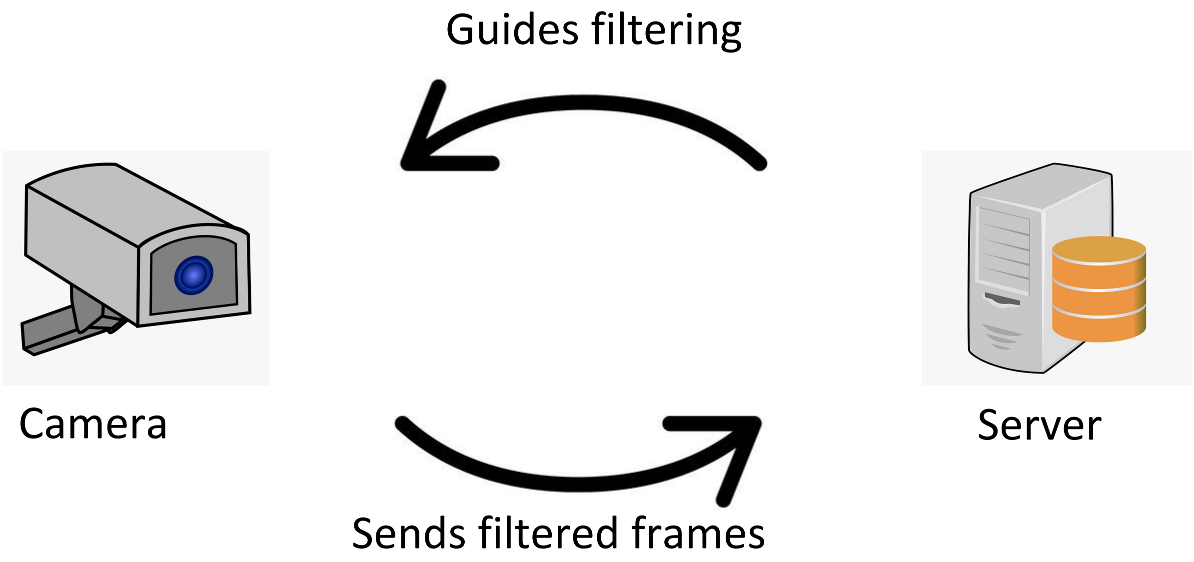
- Expand beyond pixel comparison

Pixel: 0.016
Area: 0.145

Pixel: 0.003
Area: 0.830

Reducto Overview

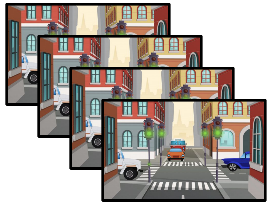
- Challenge #1: Which filtering threshold to use?
- Challenge #2: Which differencing feature to use?

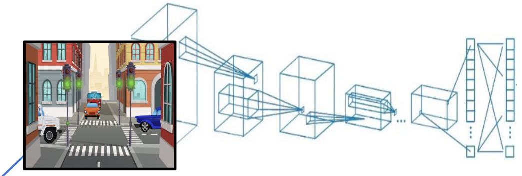


Wimpy cameras can use cheap differencing techniques to filter frames effectively with guidance from a server

Challenge #1: Threshold

- Building table is expensive -> run on server
- Looking up table is cheap -> run on camera





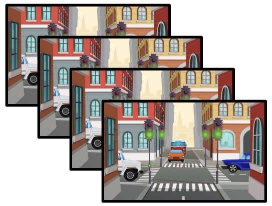
Extract differencing feature

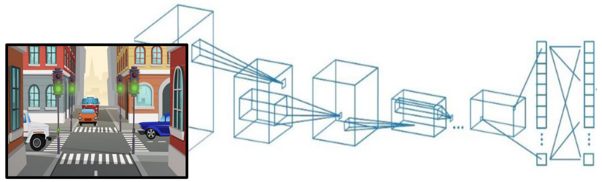
Diff	Threshold	% Filtered	Acc
0.03	…	…	…
0.04	0.01	0.15	0.98
	0.04	0.35	0.92
	~~0.07~~	~~0.60~~	~~0.88~~
	…	…	…
0.06	0.01	0.12	0.94
	~~0.04~~	~~0.30~~	~~0.89~~
	~~0.07~~	~~0.55~~	~~0.84~~
	…	…	…
0.07	…	…	…

Diff	Threshold
0.03 – 0.05	0.04
0.05 – 0.07	0.01
…	…

Challenge #2: Differencing Feature

- Calculating best feature is expensive -> run on server





% frames filtered while meeting accuracy

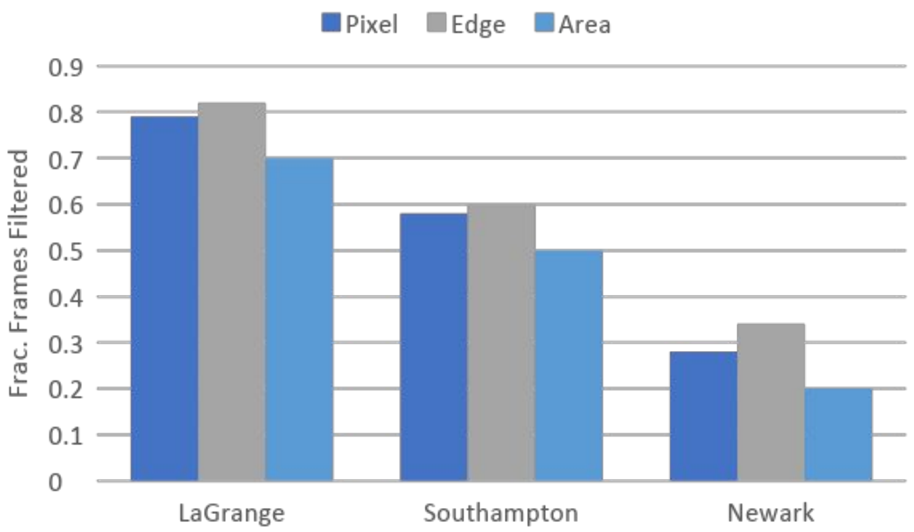
Best feature = feature that filters most frames

Extract all differencing features

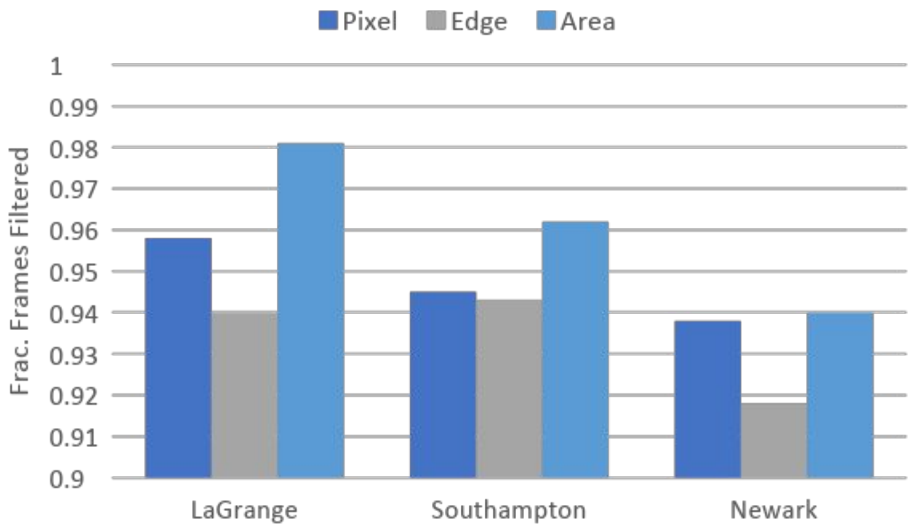
Challenge #2: Differencing Feature

- Best feature changes between query types but *not between videos*

Bounding box query

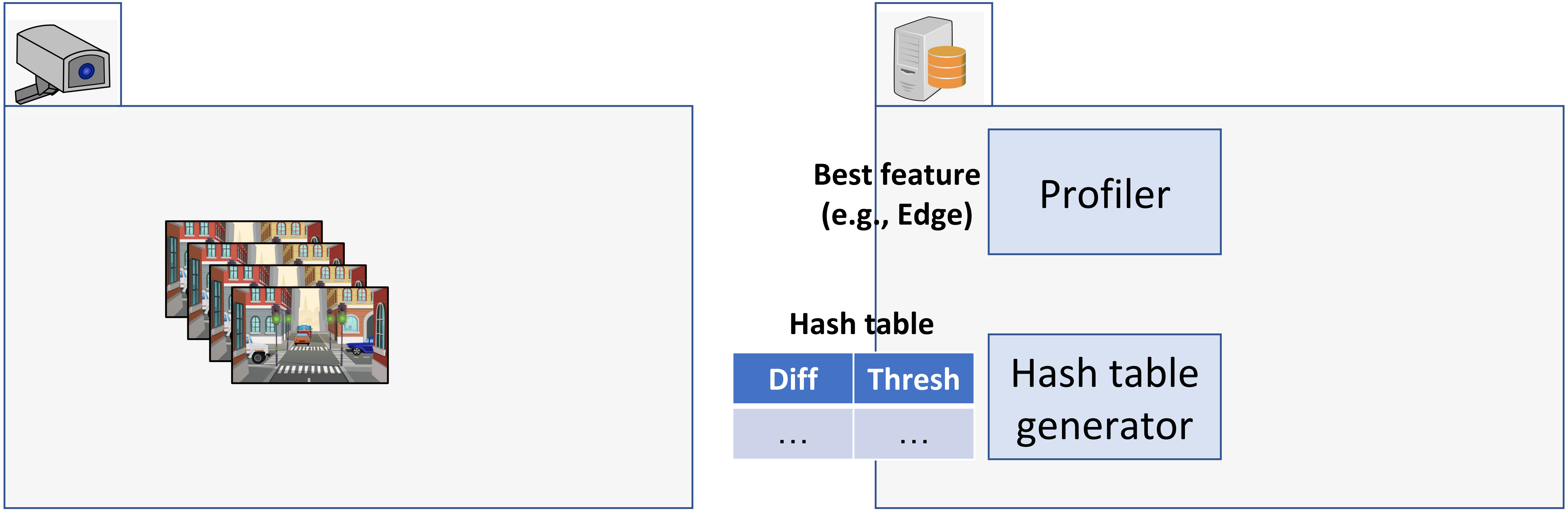


Counting query



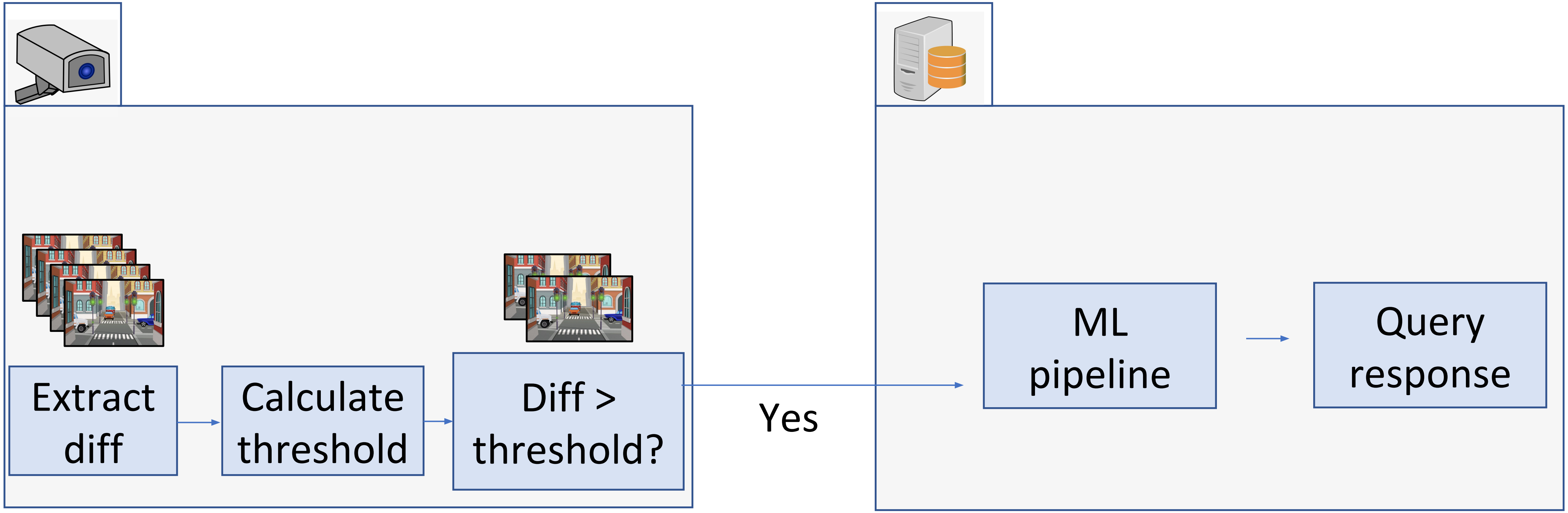
Putting It Together

Offline:



Putting It Together

Online:



Evaluation: Methodology

- Three queries: detection, counting, tagging
- 8 traffic videos: 25 10-min clips each
- DNN on server: YOLOv3
- Camera: Raspberry Pi Zero or VM with matching resources

Sample screenshots

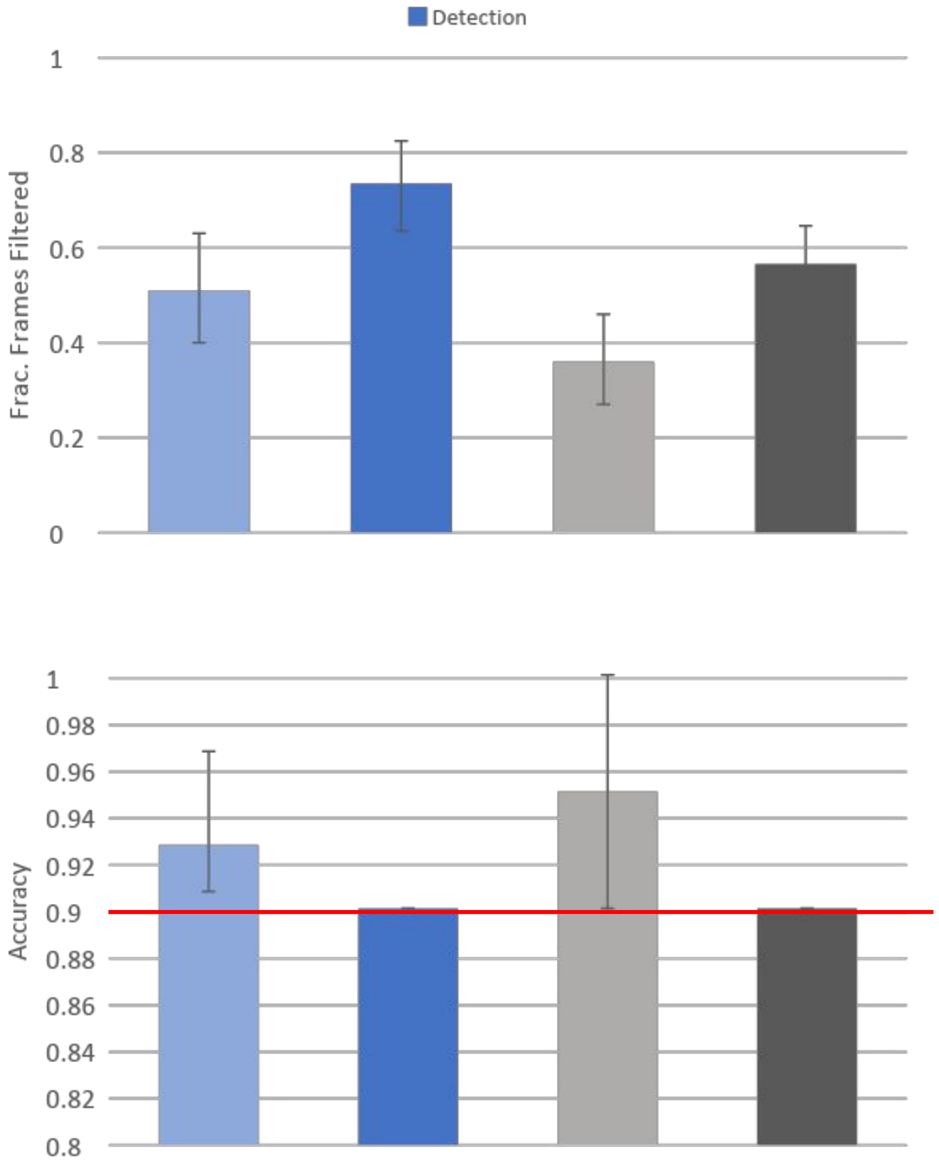
Evaluating Reducto

- **Reducto vs. offline optimal filtering**

- **Speed on Camera**

- **Compute and bandwidth savings**

Reducto vs. Offline Optimal Filtering



Reducto filters 36-51% of frames while meeting accuracy target

Speed on Camera

- 47.8 fps on Raspberry Pi Zero

Extract frame features
99.7 fps

Calculate frame difference
129.5 fps

Hash table lookups
318.6 fps

Resource Savings

Network

- Reducto saves average of 22% bandwidth

	Fraction Filtered (%)	Bandwidth Savings (%)
Baseline	0.00	0.00
Reducto	53.42	22.30
Offline	72.80	39.33

Compute

- Reducto doubles backend processing speed

	Backend processing (fps)
Baseline	41.13
Reducto	86.21
Offline	140.01

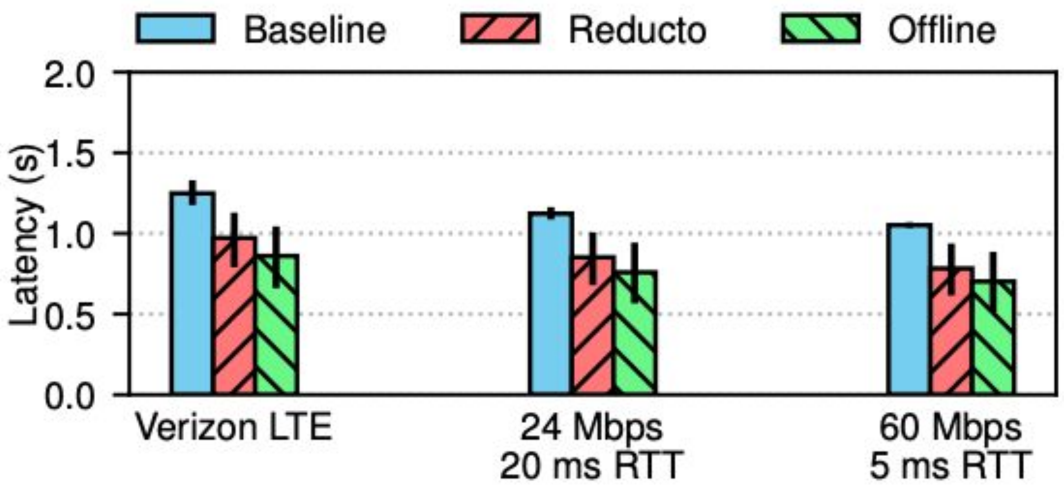
Resource Savings

Network

- Reducto saves average of 22% bandwidth

Compute

- Reducto doubles backend processing speed



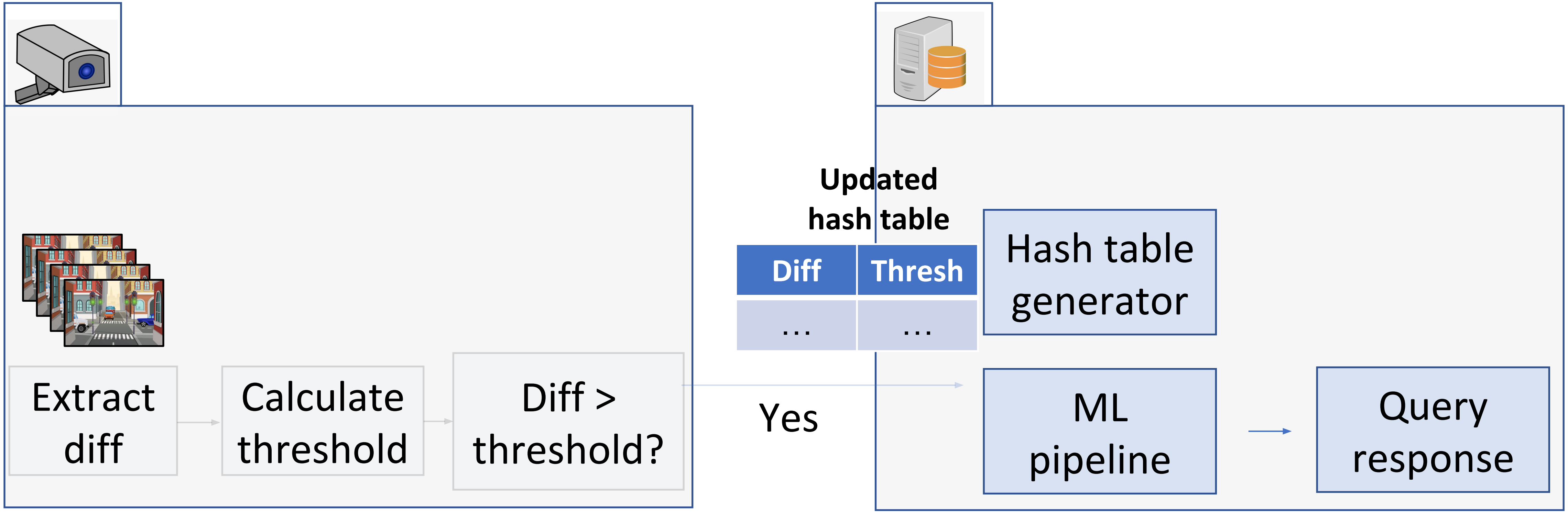
End-to-End Latency: Reduces median response time by 22-26% (within 13% of offline optimal)

Comments

- **Pros:**
  - Insightful observation & significant performance.
  - Very good writings. Explain the design choices well.
- **Cons:**
  - This work is based on “frame filter” types of work. The idea itself is not very novel.
- **Takeaway:**
  - Use comprehensive data and survey to support motivation & observations.
  - Good Explanation for design choice & observations makes good paper.

Putting It Together

Online:



Putting It Together

Online: