Democratizing Video Analytics – The quest for the *holy trinity* of low latency, low cost, and high accuracy

Ganesh Ananthanarayanan

Azure for Operators

http://aka.ms/ganesh

Microsoft

Ganesh Ananthanarayanan

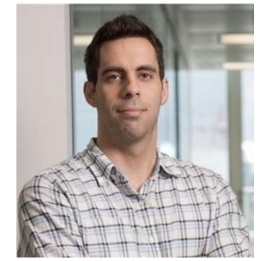
Victor Bahl

Peter Bodik

Yuanchao Shu

Shivaram Venkataraman

Junchen Jiang



Stavros Volos

Michael Hung

Kevin Hsieh

Haoyu Zhang

Samvit Jain

Rishabh Poddar

Enrique Saurez

Leana Golubchik

Minlan Yu

Michael Freedman

Phil Gibbons

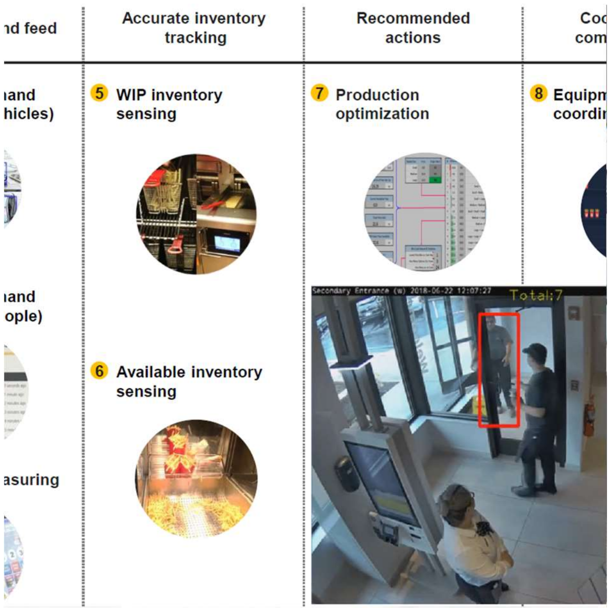
Ion Stoica

Raluca Popa

Onur Mutlu

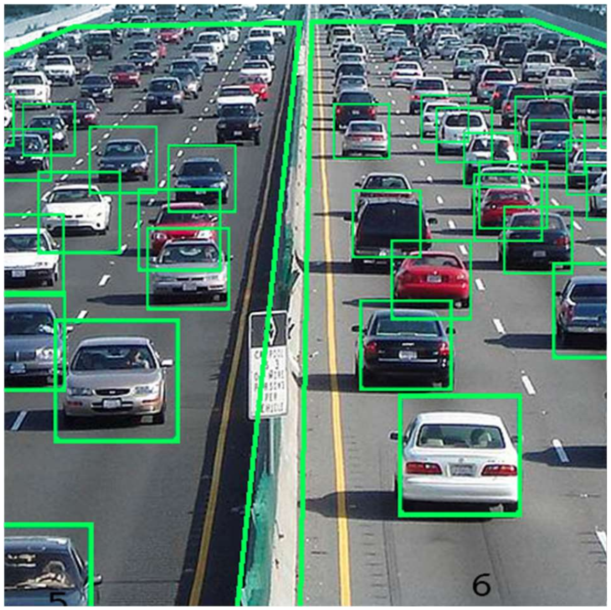
Joey Gonzalez

Cameras are everywhere!



Connected Restaurants

Retail Stores



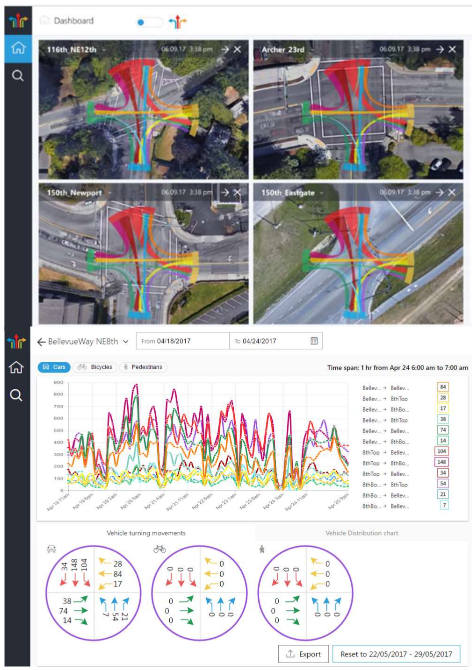
Smart Cities & Urban Mobility

Video analytics & real-time actuation is integral to the promise of 5G

[1] Smart city video analytics on 5G edge hierarchy

Car/bike/pedestrian counts & near-collisions by analyzing widely-deployed traffic cameras

Dashboard & alerts

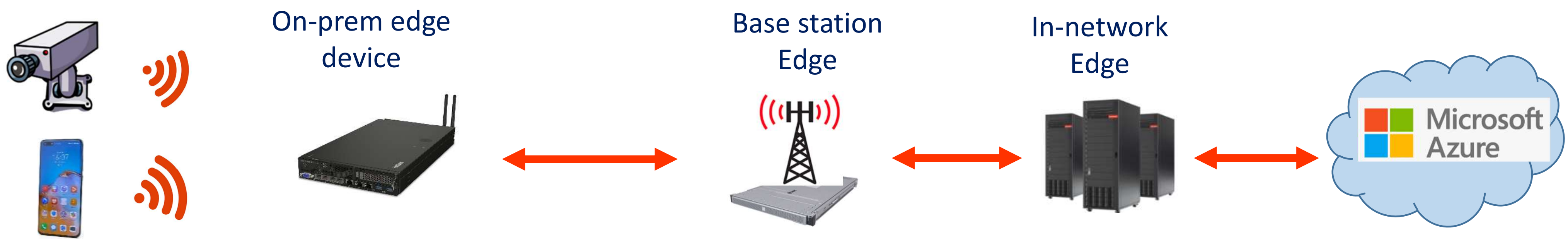


Analytics & actuation

(***Built up on prior work with City of Bellevue***)

[1] Smart city video analytics on 5G edge hierarchy

Vehicle counts over hierarchy of edges in 5G infrastructure

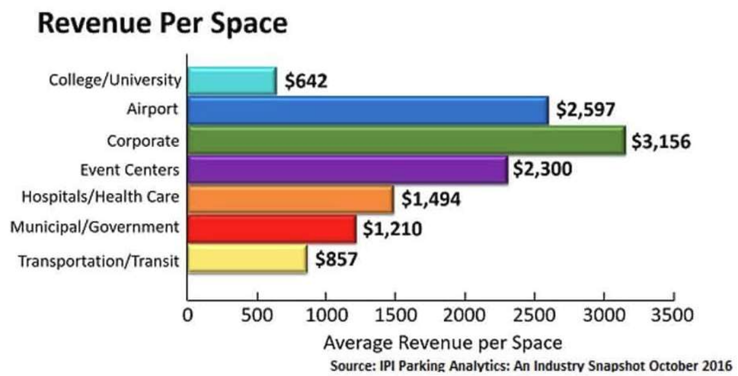


- ✓ Six-fold reduction in network traffic between the edges in the hierarchy, thus lowering the bandwidth needed to be provisioned
- ✓ Reduction in compute provisioning of edge devices via smart placement
- ✓ Vehicle counts from traffic camera videos with nearly 100% accuracy

[2] 5G Parking Services with Edge Compute

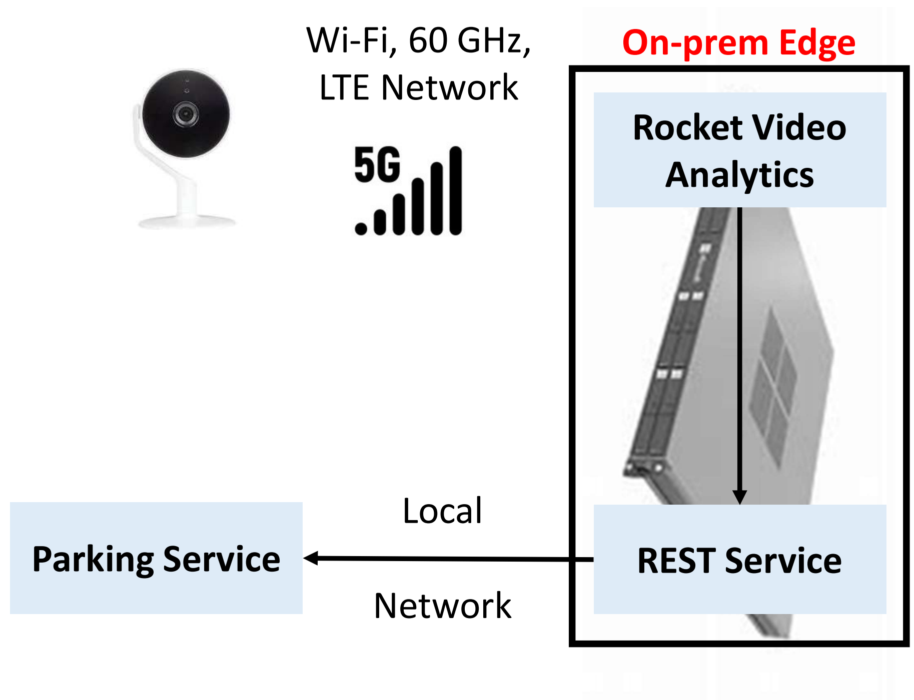
FUJITSU

Parking Application: Finding parking can increase stress associated with traveling, CO2 emission, and traffic congestion (driving in circles)



Sensors vs. Cameras

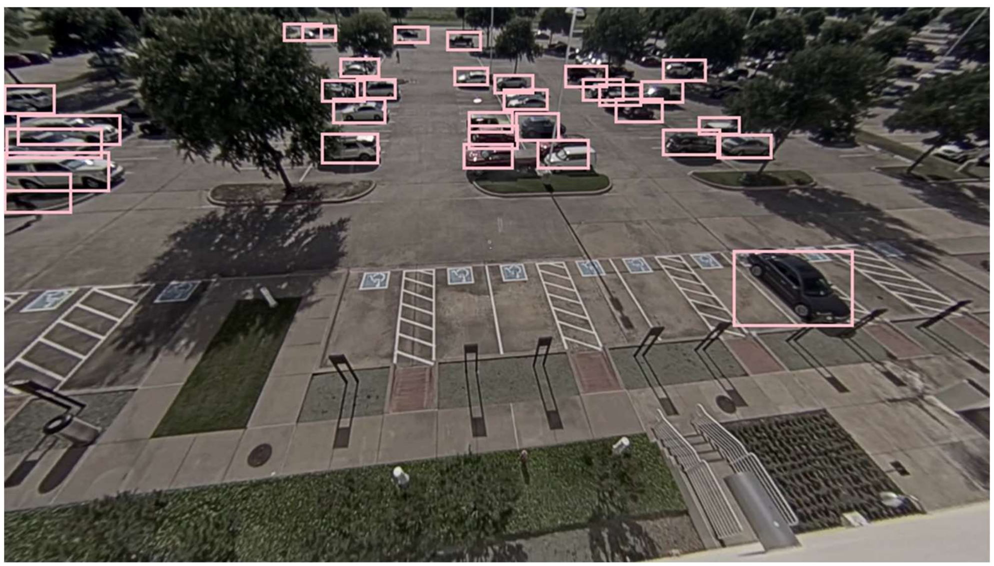
- ✓ Easily extend to other applications
- ✓ Cheap to scale up



[2] 5G Parking Services with Edge Compute

FUJITSU

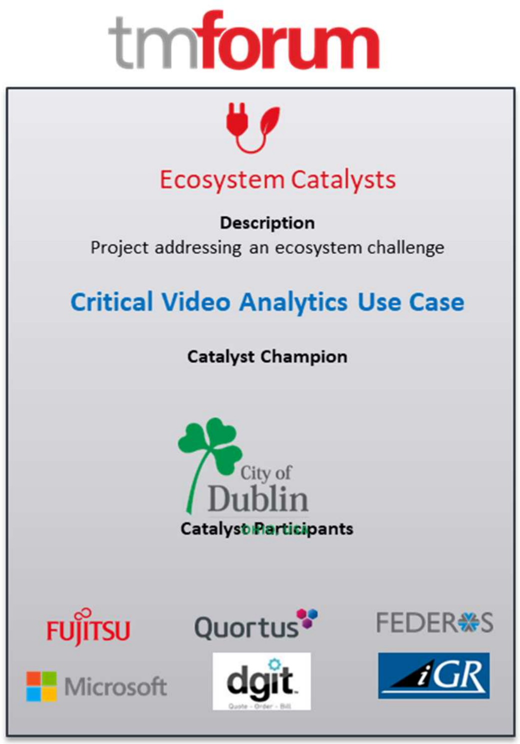
Parking Application: Finding parking can increase stress associated with traveling, CO2 emission, and traffic congestion (driving in circles)

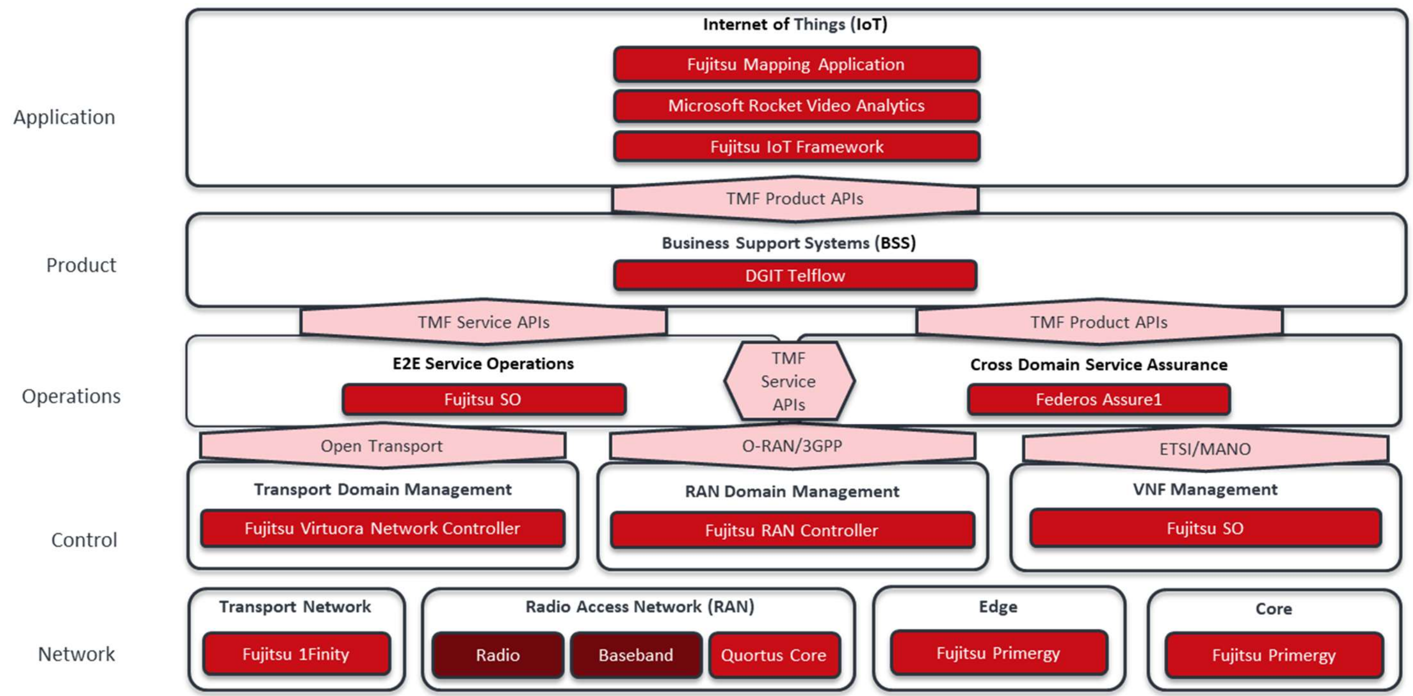


Analyze live videos → detect vehicles → infer occupancies

[2] 5G Parking Services with Edge Compute

Parking Application: Finding parking can increase stress associated with traveling, CO2 emission, and traffic congestion (driving in circles)





Democratize live video analytics!

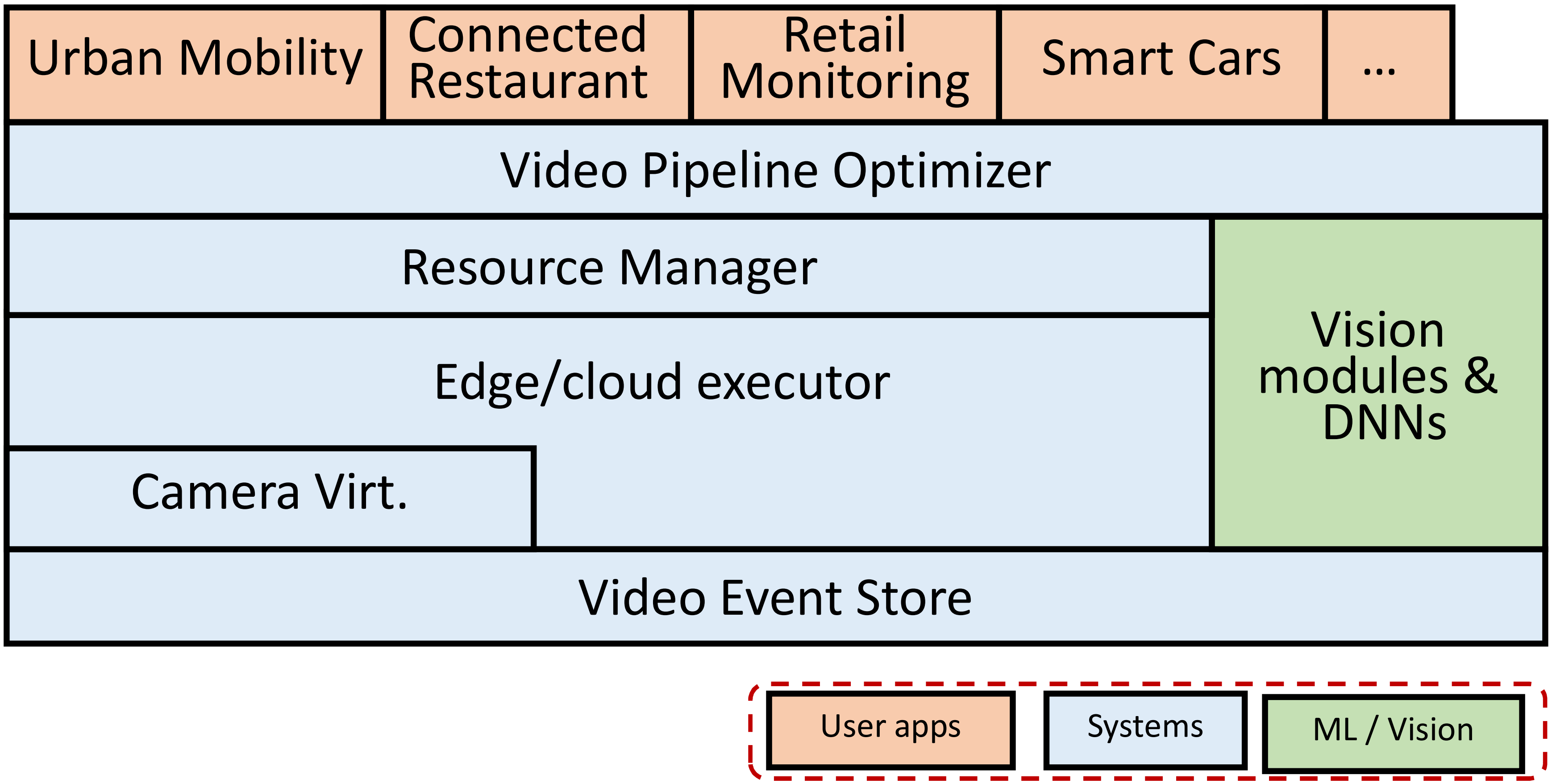
low-cost, accurate, private
video analytics system
for a collection of cameras

"Real-time Video Analytics – the killer app for edge computing", IEEE Computer 2017

Because of the high data volumes, compute demands, and latency requirements, we believe that cameras represent the most challenging of "things" in Internet-of-Things

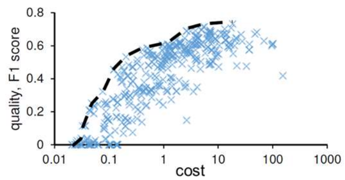
Rocket: Video Analytics Stack

http://aka.ms/rocket

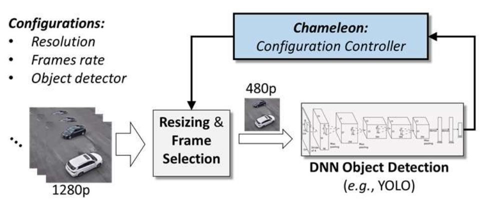


This talk will cover…

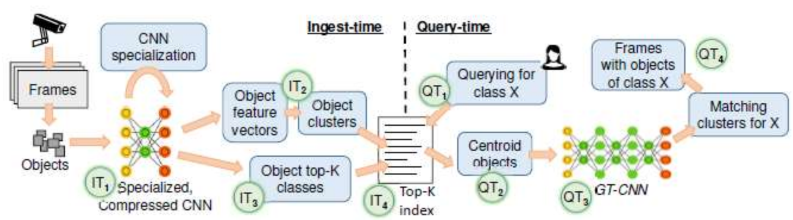
- Video analytics pipelines across edge/cloud with *approximation*



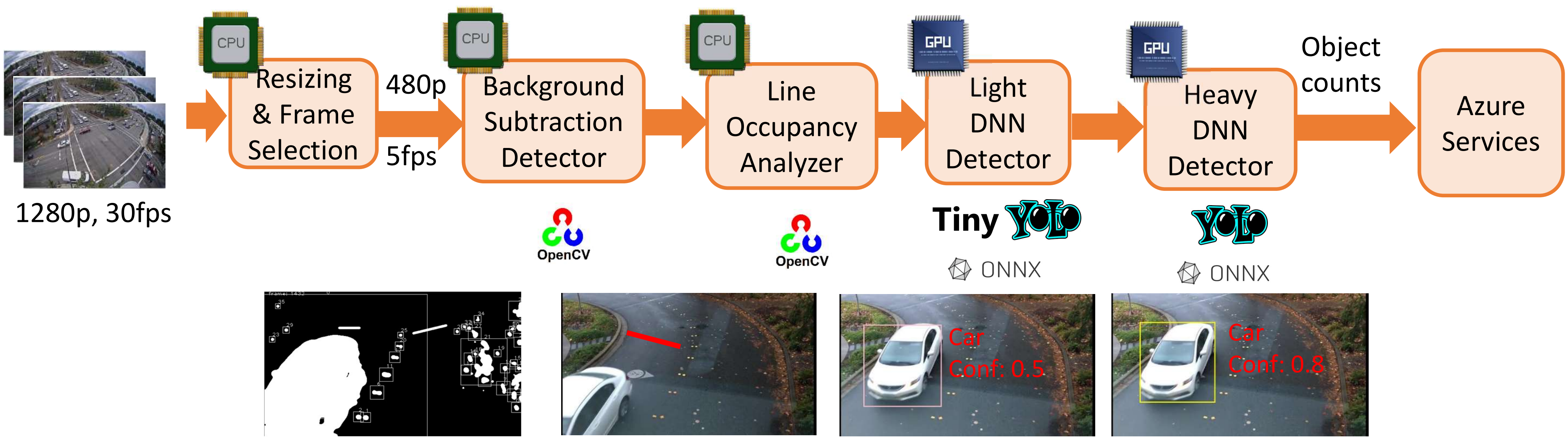
- Adaptive video analytics at scale



- Interactive querying of stored video datasets



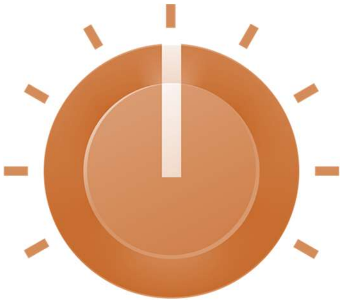
Cascaded video analytics pipeline



Configurations:

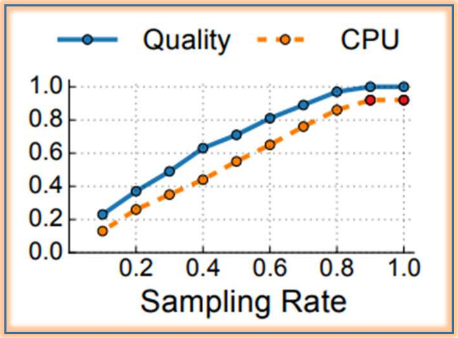
- *Resolution*
- *Frames rate*
- *Object detector*

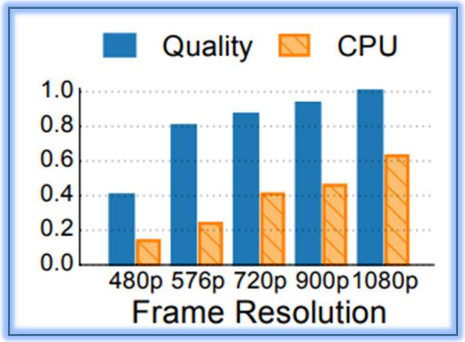
Video analytics pipelines



Frame Rate

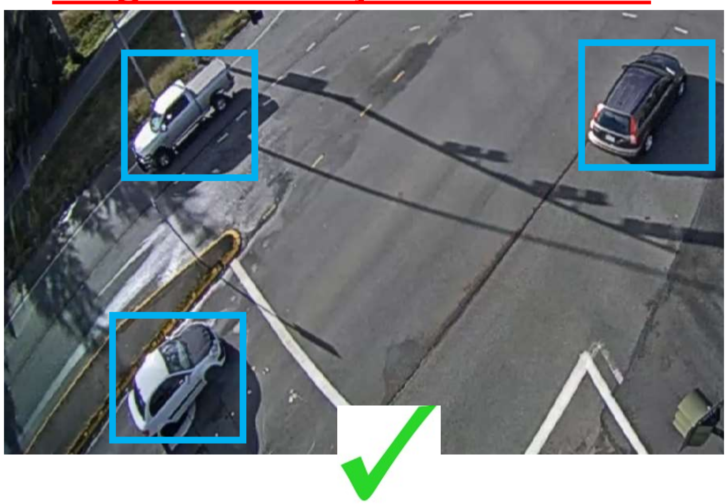
Resolution



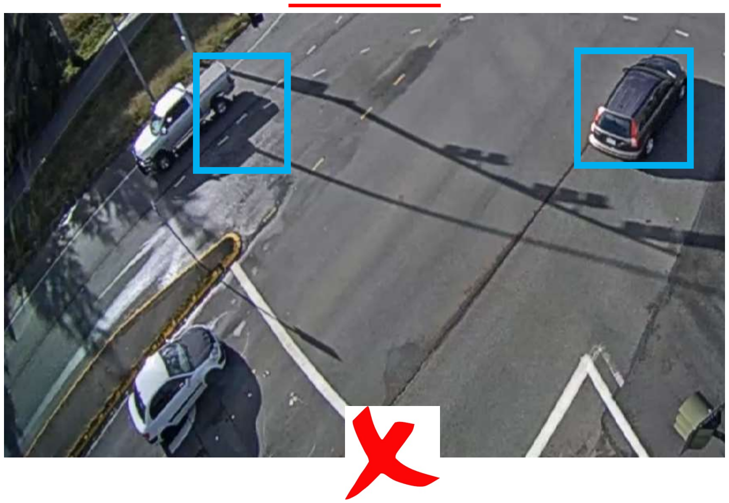


Video analytics pipelines

Single Shot object detection



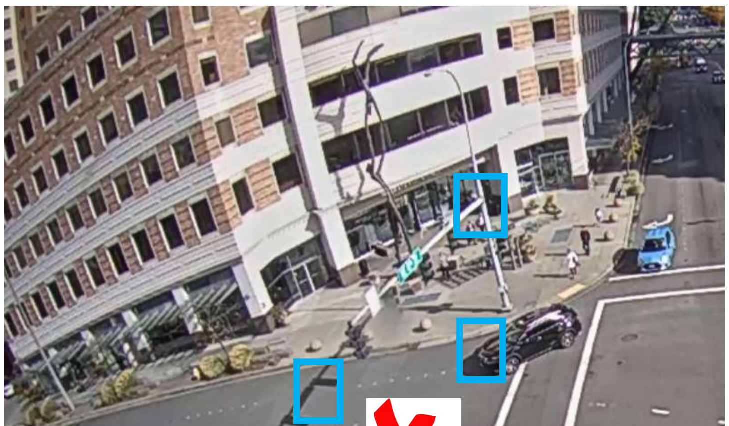
Yolo v3



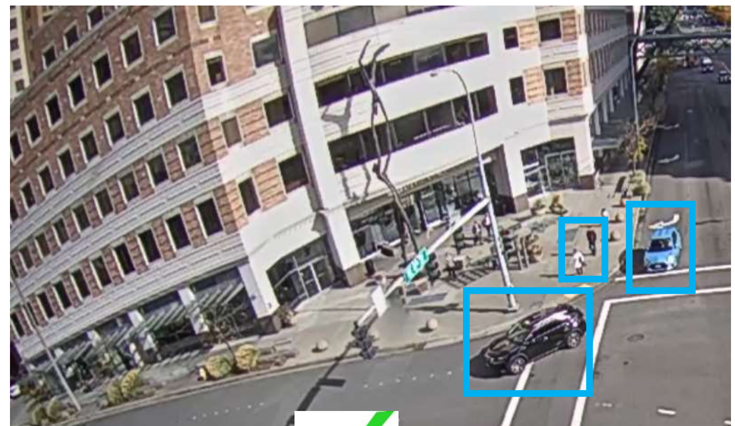
150th NE and Newport Ave
Bellevue, WA

Video analytics pipelines

Single Shot object detection

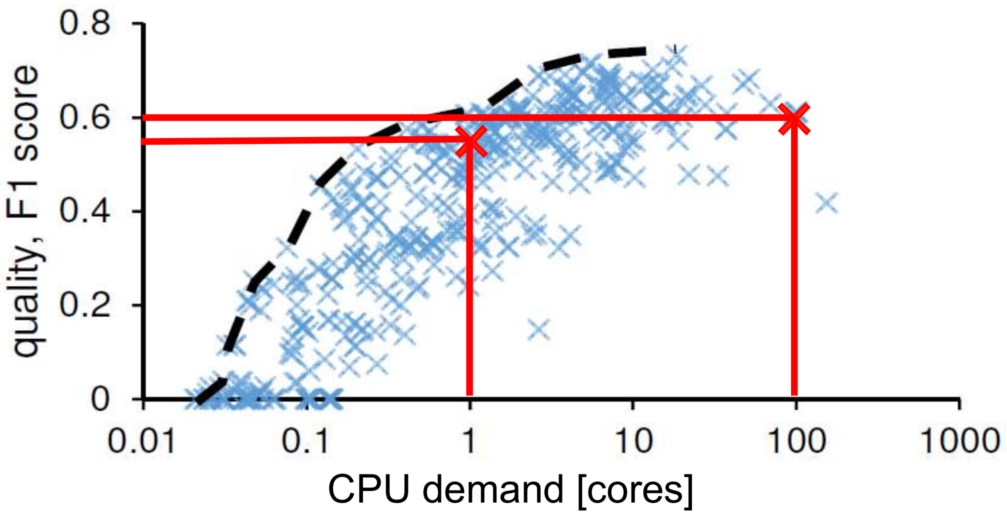


Yolo v3



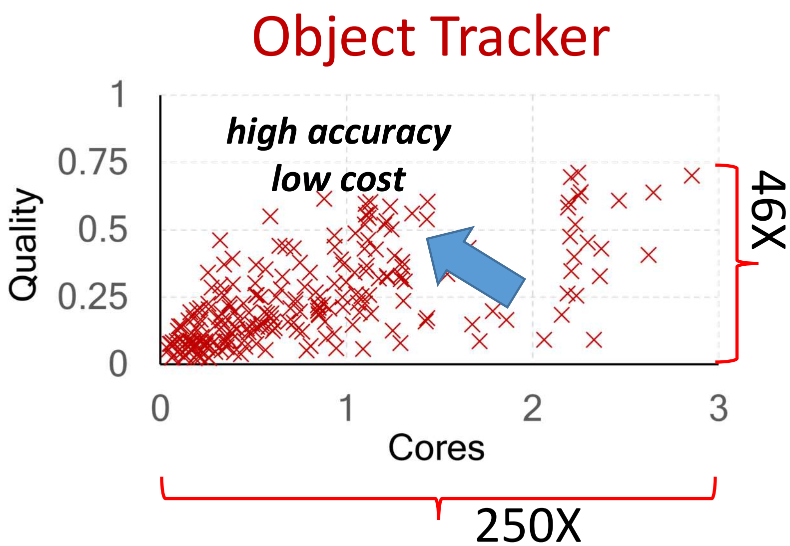
Bellevue Ave and NE 8th
Bellevue, WA

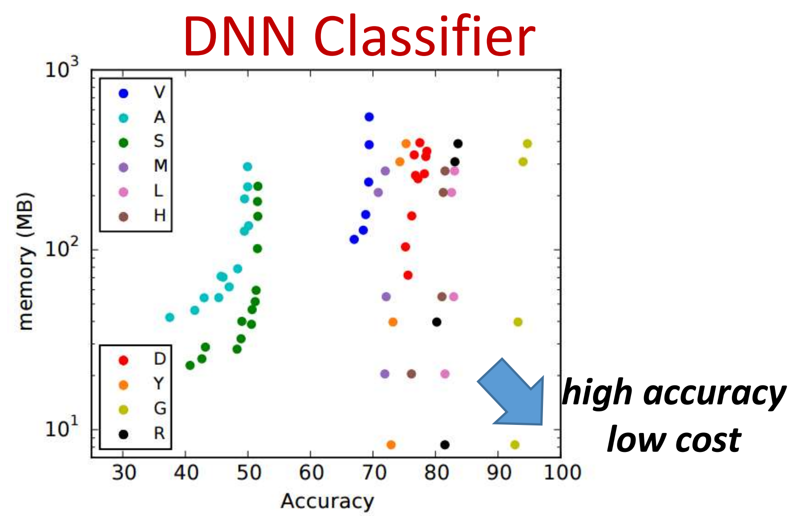
How much do the *configurations – knobs & implementations –* differ?



Orders of magnitude cheaper resource demand for little quality drop

How much do the *configurations – knobs & implementations* – differ?



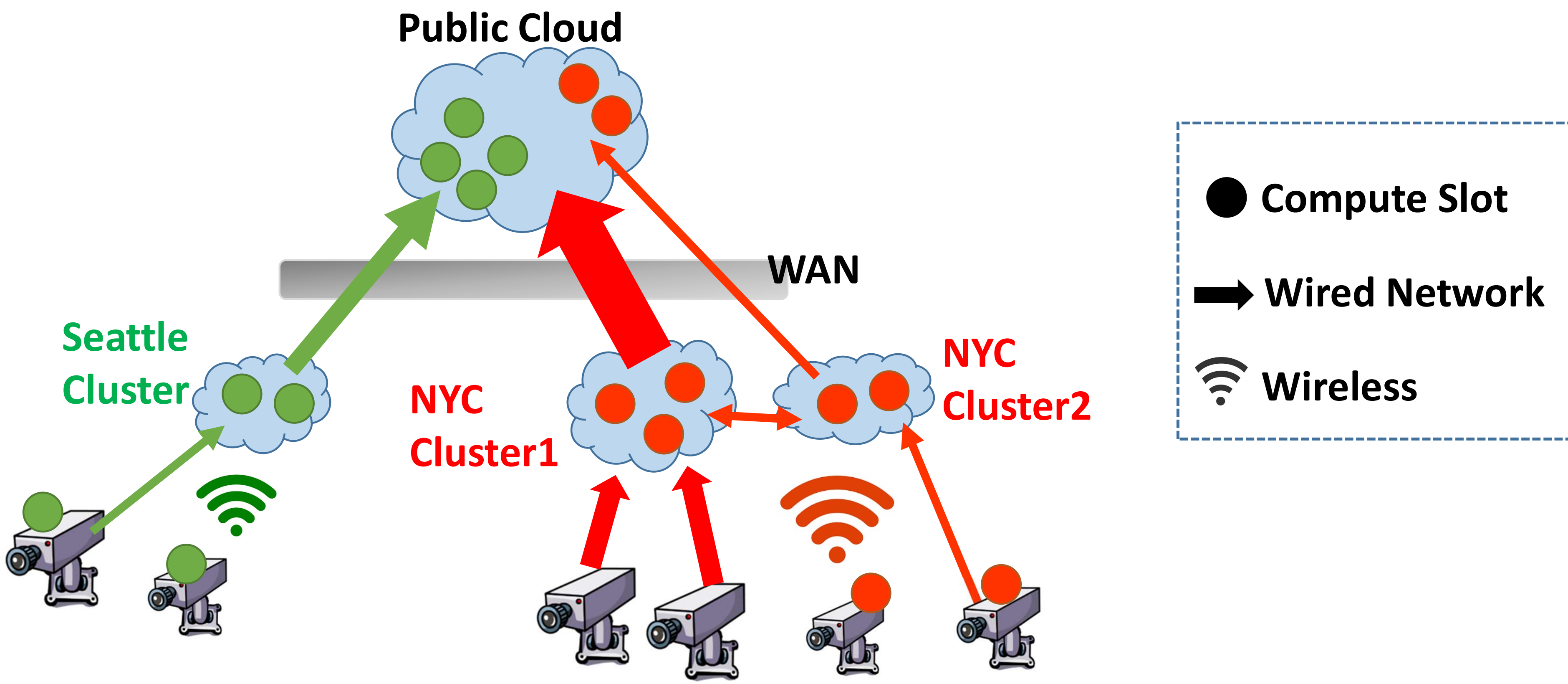


Dependent on the camera, lighting, object color, …

No analytical models to construct resource-quality profiles

- Different from SQL queries

Hierarchy of clusters for video analytics

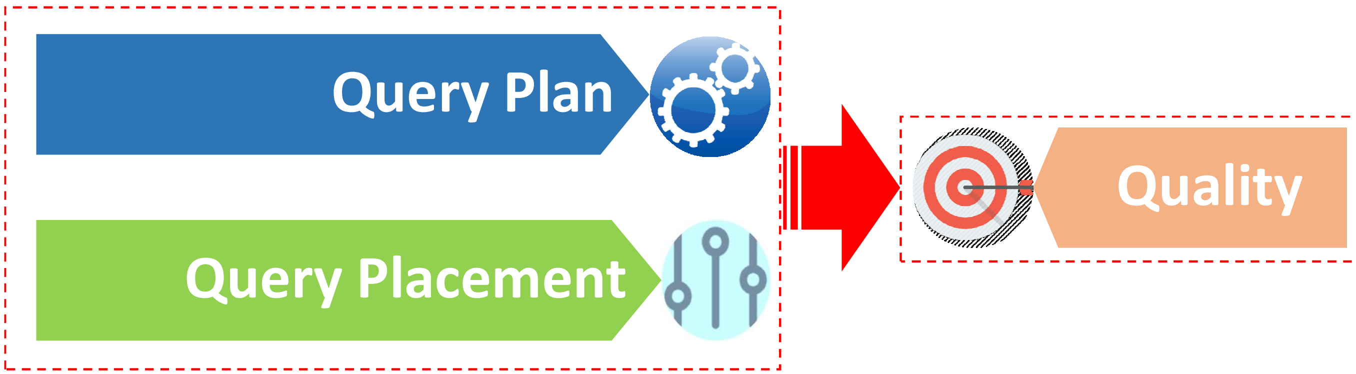


1. Pick the *configurations – knobs & implementations –* for video queries

(jointly)

2. Place the modules across the hierarchy of clusters

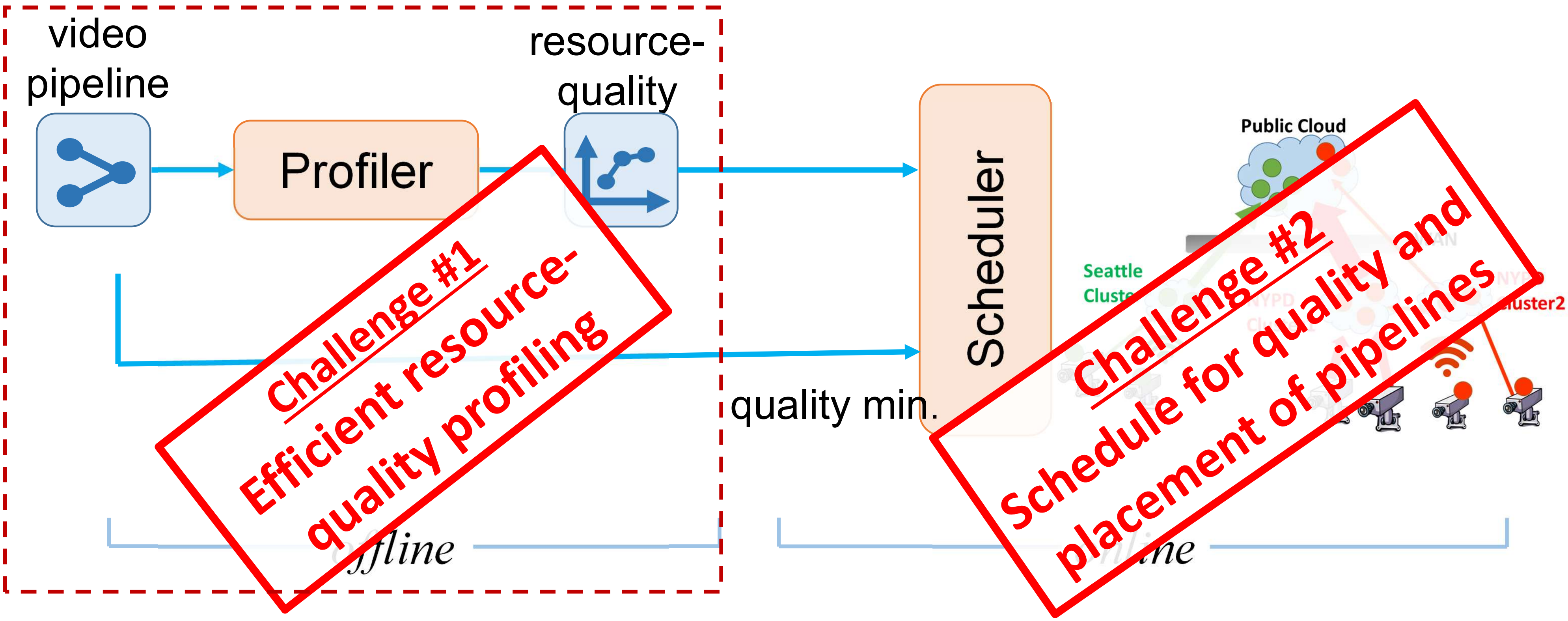
Decide **configurations** and **placement** to maximize **quality** *across multiple video pipelines* within the hierarchical resource capacity



Diverse Quality Requirements

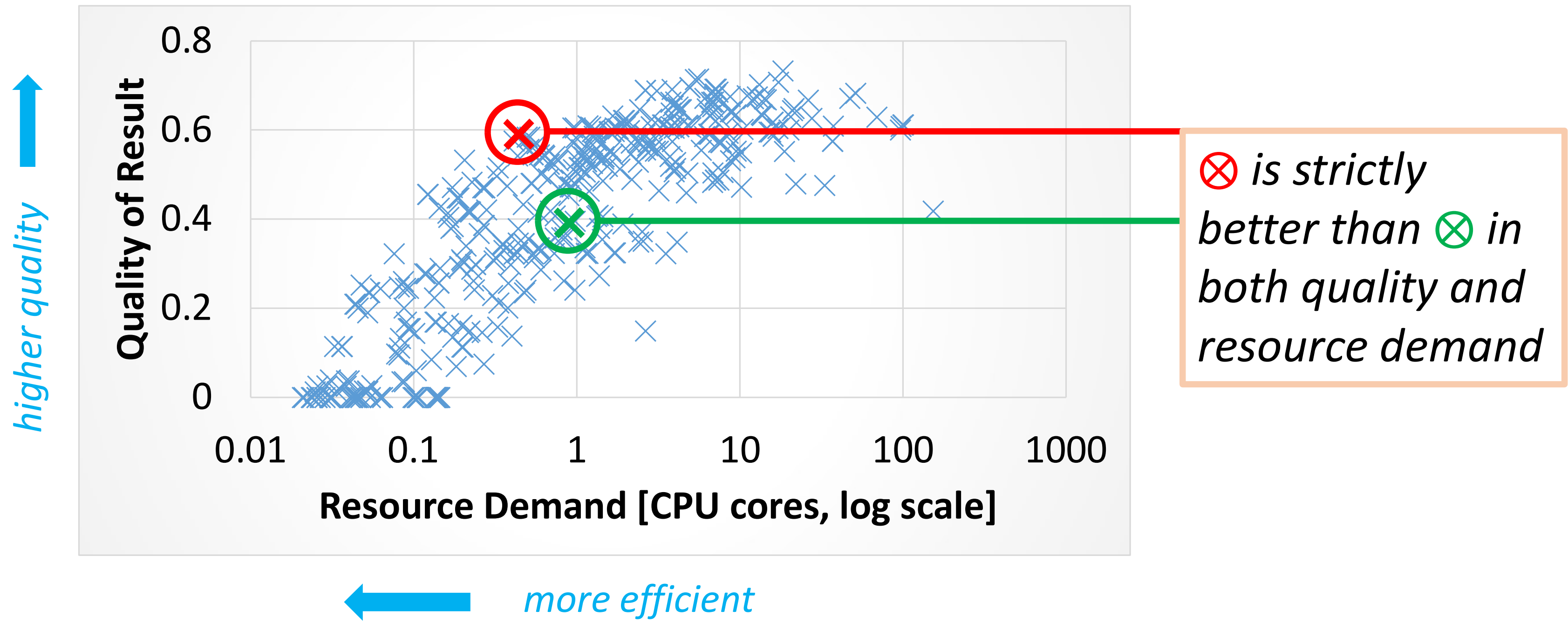
Applications can set their minimum quality

Solution Overview



Offline: Resource-Quality Profiling

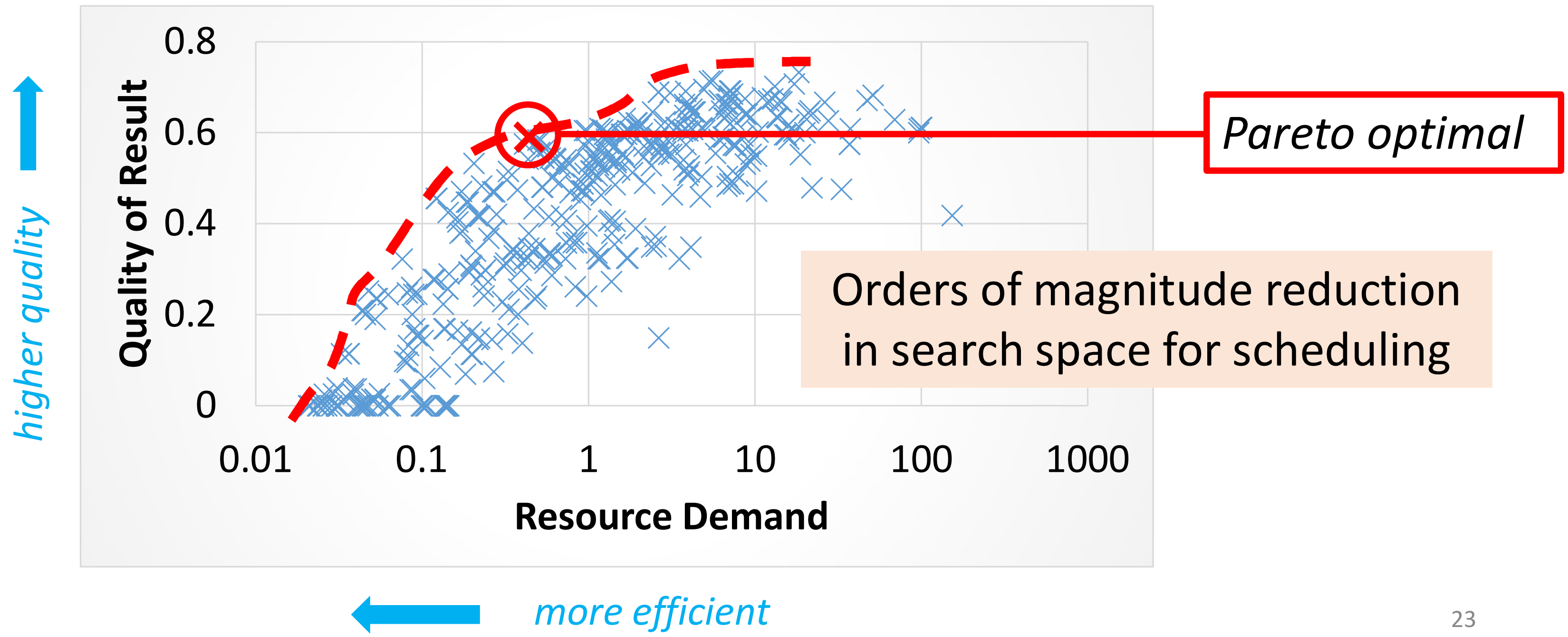
- Profile: configuration ⟹ {resource, quality}
  - <u>Ground-truth:</u> labeled dataset or results from *golden* configuration
  - Targeted search for promising configurations



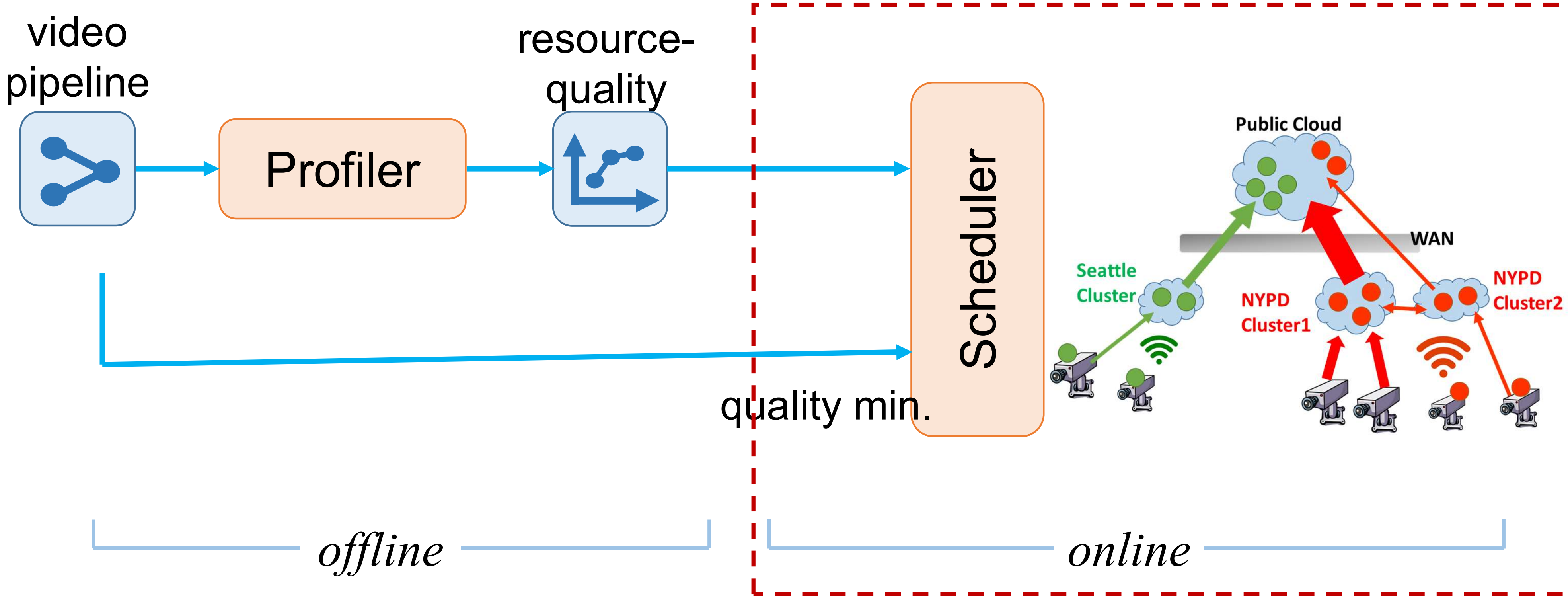
Offline: Pareto boundary

Pareto boundary: optimal configurations in resource demand and quality

- Non-Pareto plans cannot beat Pareto configs. in *both* quality & resources

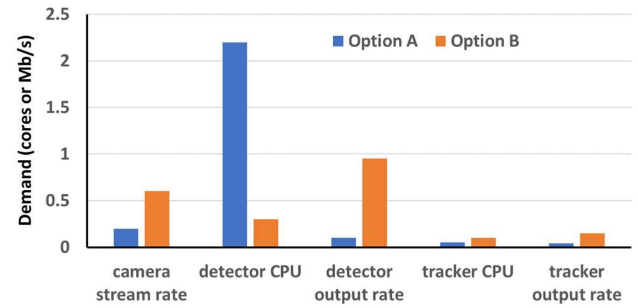


Solution Overview



Scheduling Heuristic

A and B have accuracy of 0.74



Dominant Resource Demand

- Multi-resource – compute & network
- For each (configuration, placement) pair, calculate the *fraction* of demand at *each location*

  → calculate the max (or dominant) fraction

✓ Avoids lopsided drain of any single resource at any location

✓ Dimensionless – extends to multiple resource types

"VideoEdge: Processing Camera Streams using Hierarchical Clusters", ACM SEC 2018

Scheduling Heuristic

Greedy Allocation

- Dominant demands of all (configuration, placement) options
- Allocate in small increments of dominant resource
- Prefer options whose (improvement in accuracy / additional resource) is highest

- Optimizes for *average accuracy* of video pipelines

✓ Merge common modules across pipelines
  - E.g., Two pipelines analyzing the same video stream can share their object detector DNNs

"VideoEdge: Processing Camera Streams using Hierarchical Clusters", ACM SEC 2018

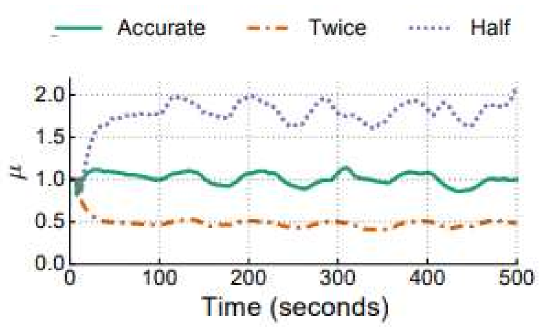
Evaluation Highlights

Workload

- Videos from traffic cameras & surveillance cameras
  - Original frame rate of 14 – 30 fps, resolution 480p – 1080p
- Workload: Object tracker, DNN classifier, Car counter, License plate reader

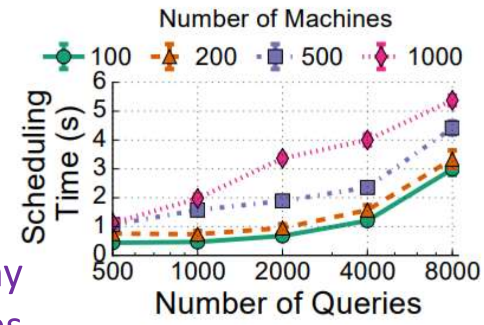
Results

- 25x better accuracy & within 6% of optimal



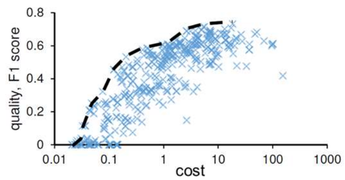
Adapts to errors in the profile

Scales to many 1000's of queries

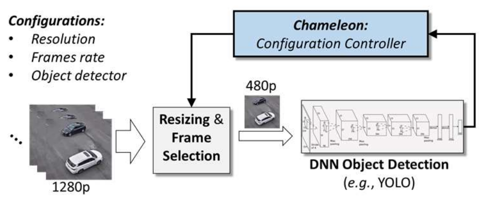


This talk will cover…

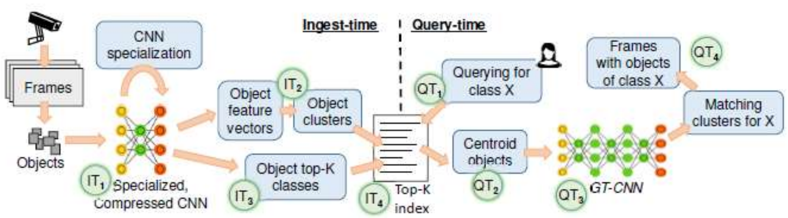
Video analytics pipelines across edge/cloud with *approximation*



- Adaptive video analytics at scale

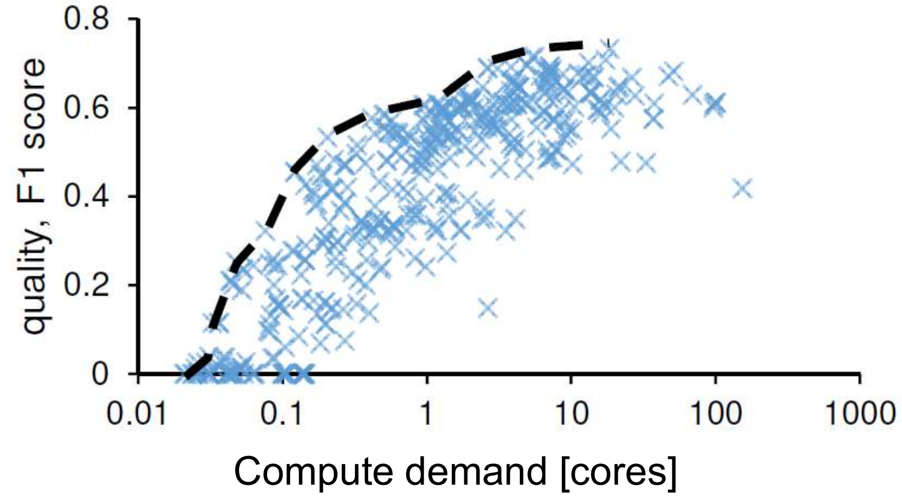


- Interactive querying of stored video datasets



Customize the video pipeline to the video content

➔Pick the best configuration by profiling *at beginning*
➔Record sample videos for resource-accuracy profile

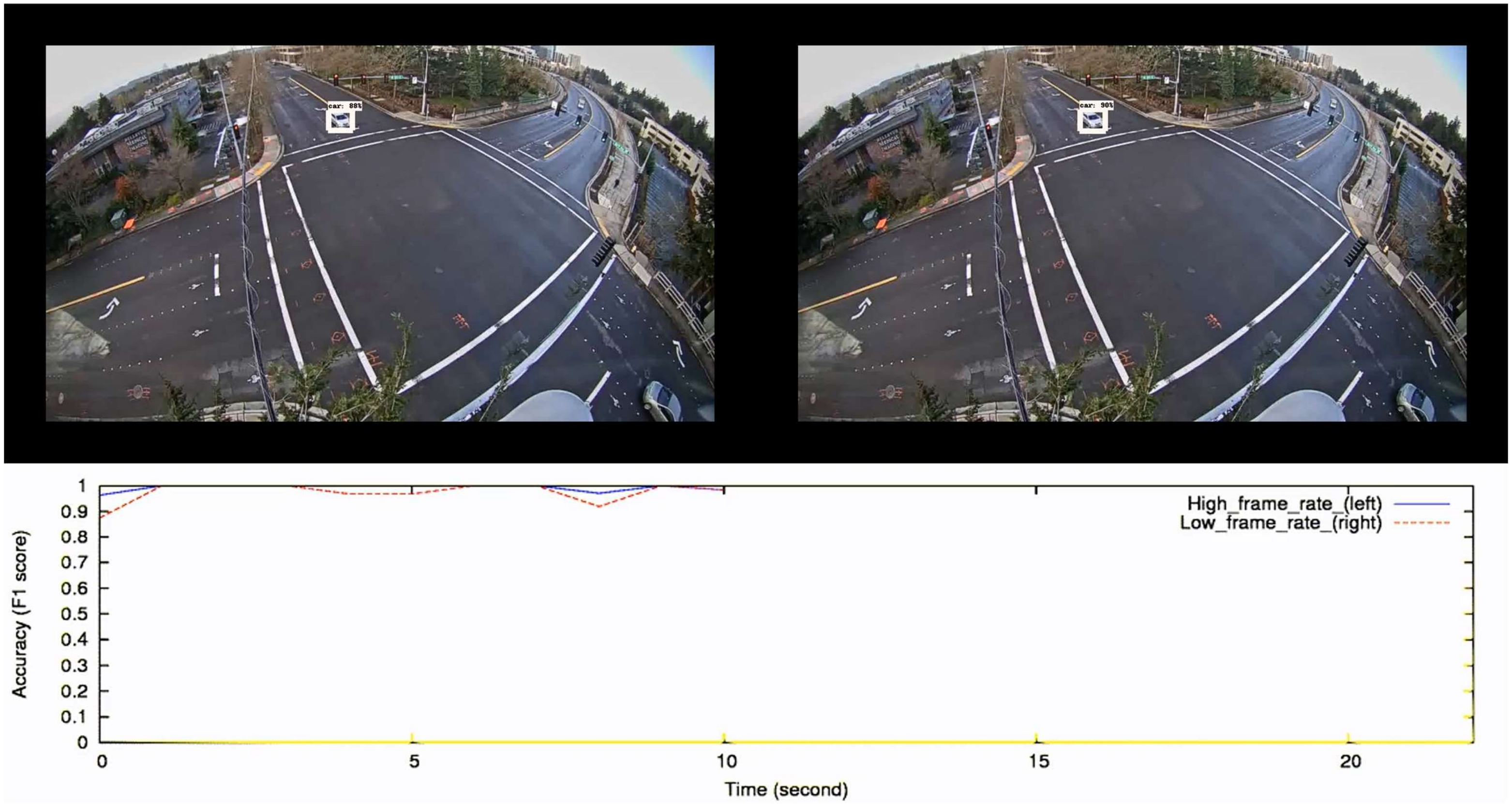


VideoStorm [NSDI 2017]
NoScope [VLDB 2018]

Frame rate


High_frame_rate_(left)
Low_frame_rate_(right)
Accuracy (F1 score)
Time (second)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
5
10
15
20

Best frame rate depends on content



Key observation

Video content varies over time

$\Rightarrow$ best configuration varies over time

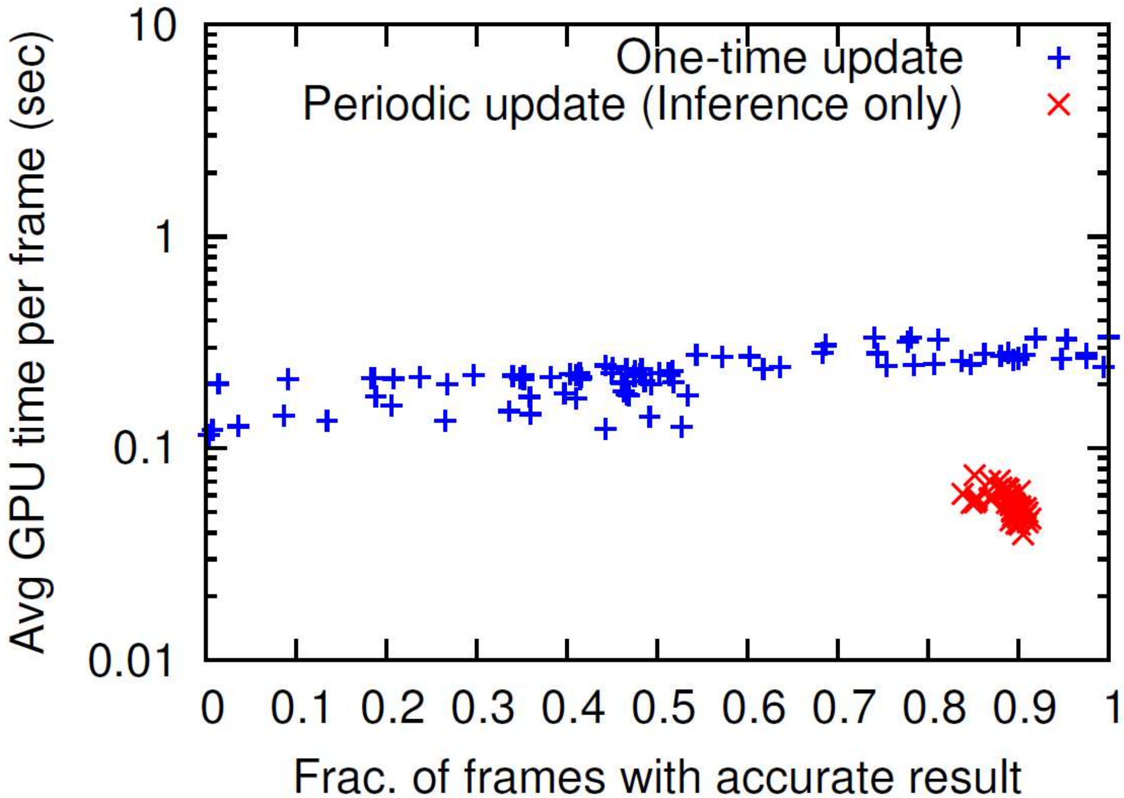
- Holds for other configuration knobs (resolution, NN classifier, etc.)

- ~~One-time profiling at beginning will not cut it!~~

~~Customize~~ Adapt the video pipeline to the dynamic video content

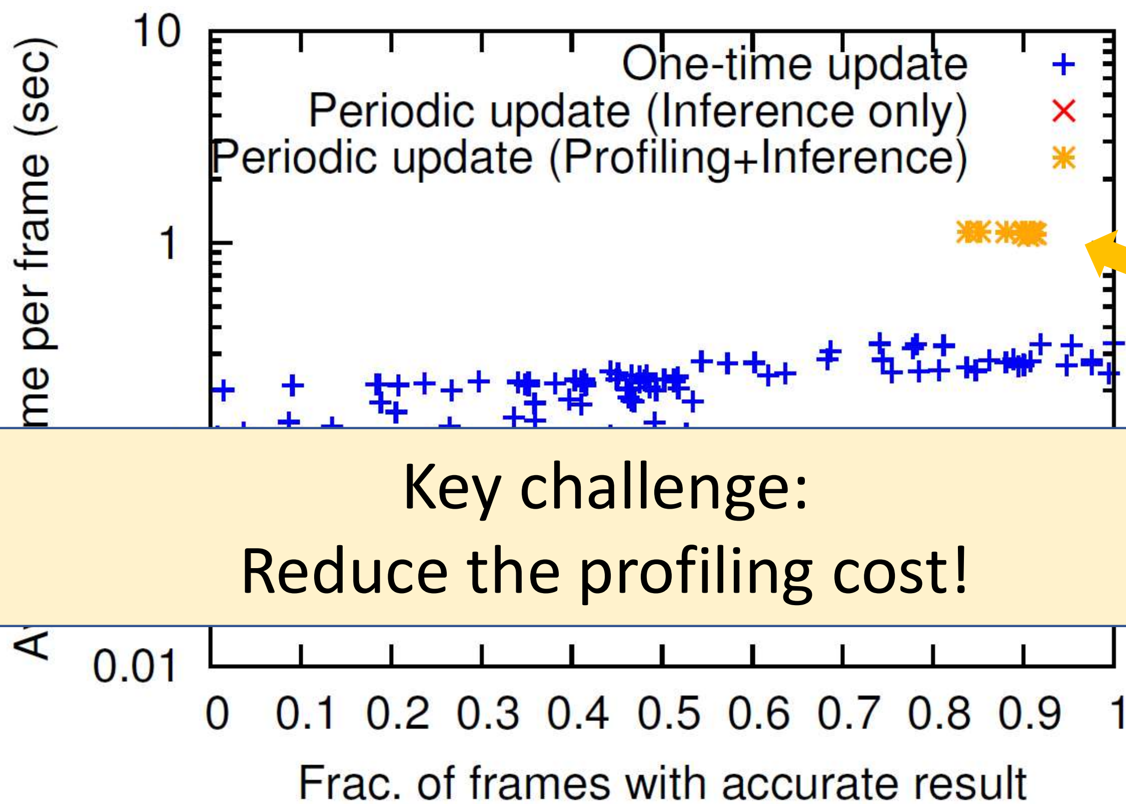
Adapt dynamic

~~Customize~~ the video pipeline to the video content



Potential win…

Adapt ~~Customize~~ the video pipeline to the dynamic video content



Potential win…
but at what cost?

Key challenge:
Reduce the profiling cost!

Adapt
~~Customize~~ the video pipeline to the dynamic video content

Key challenge:
Reduce the profiling cost!

Idea #1: Temporal correlation
Idea #2: Spatial (cross-camera) correlation
Idea #3: Independence of configurations

Idea #1: Temporal correlation

- **Insight:** Underlying characteristics of video remain stable for short periods of time
  - E.g.: size/class of objects, viewing angle

- Good configurations tend to remain good for a short while
- Bad configurations tend to remain bad for a long time!

Idea #2: Spatial cross-camera correlation

- **Insight:** Many cameras feeds share similar characteristics
  - E.g.: traffic cameras in a city see similar vehicles, weather, and viewing angles (thanks to uniform installation policies)

- Good/bad configurations for one camera tend to be good/bad for other cameras

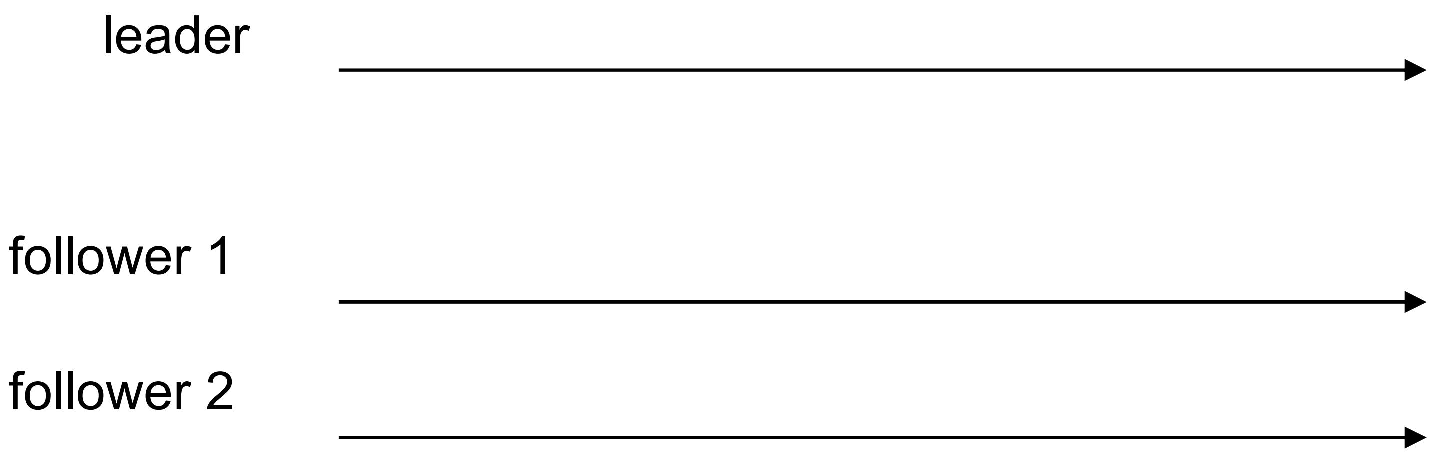
- How to find groups of similar cameras?
  - Current solution: simple offline clustering (built upon k-means)

A couple caveats …

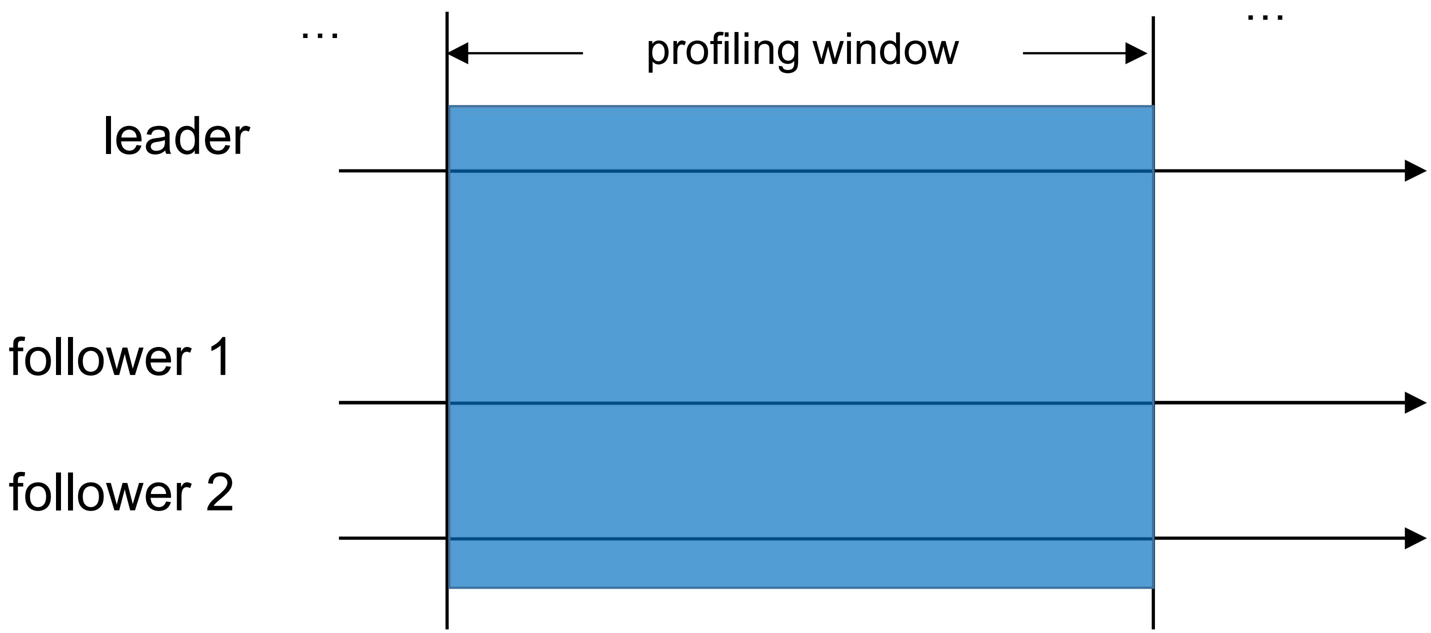
1. Applying the single best configuration temporally/spatially is unstable
   - But top-*k* configurations are more stable

2. Correlations do not hold indefinitely
   - Must periodically explore the full configuration space

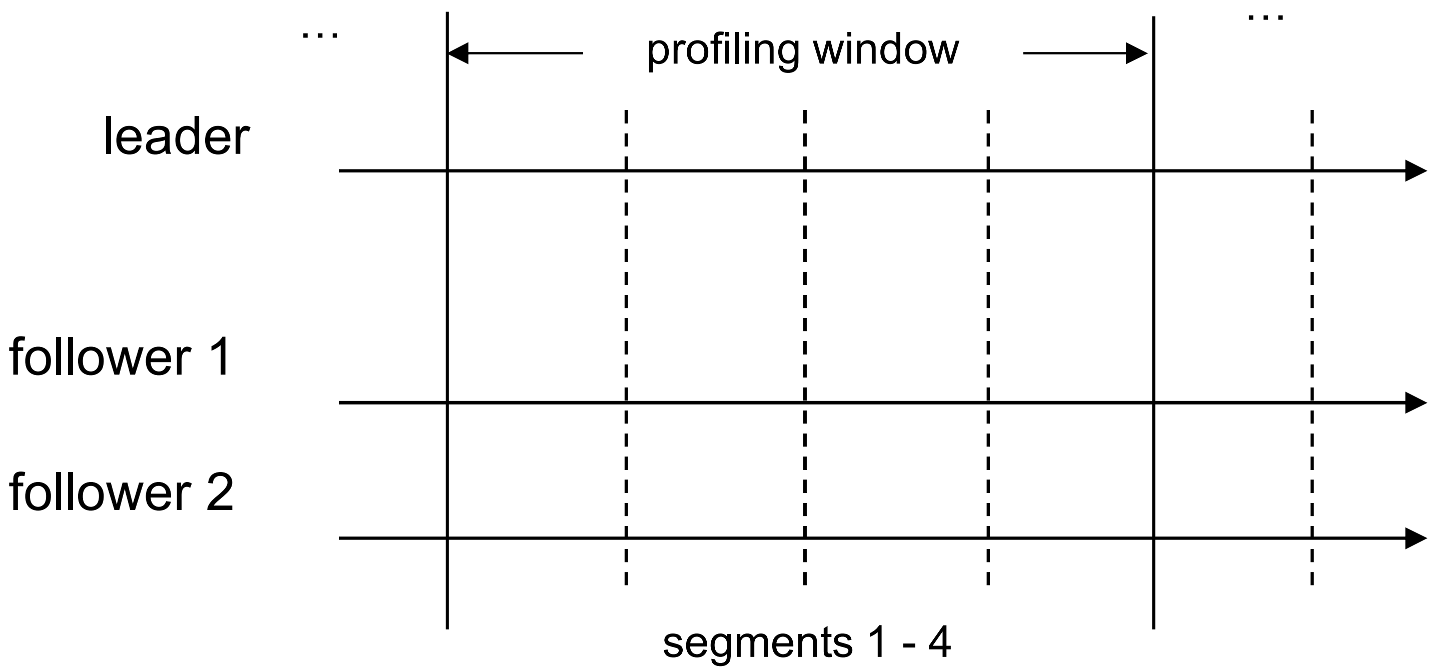
Putting them together (Chameleon)



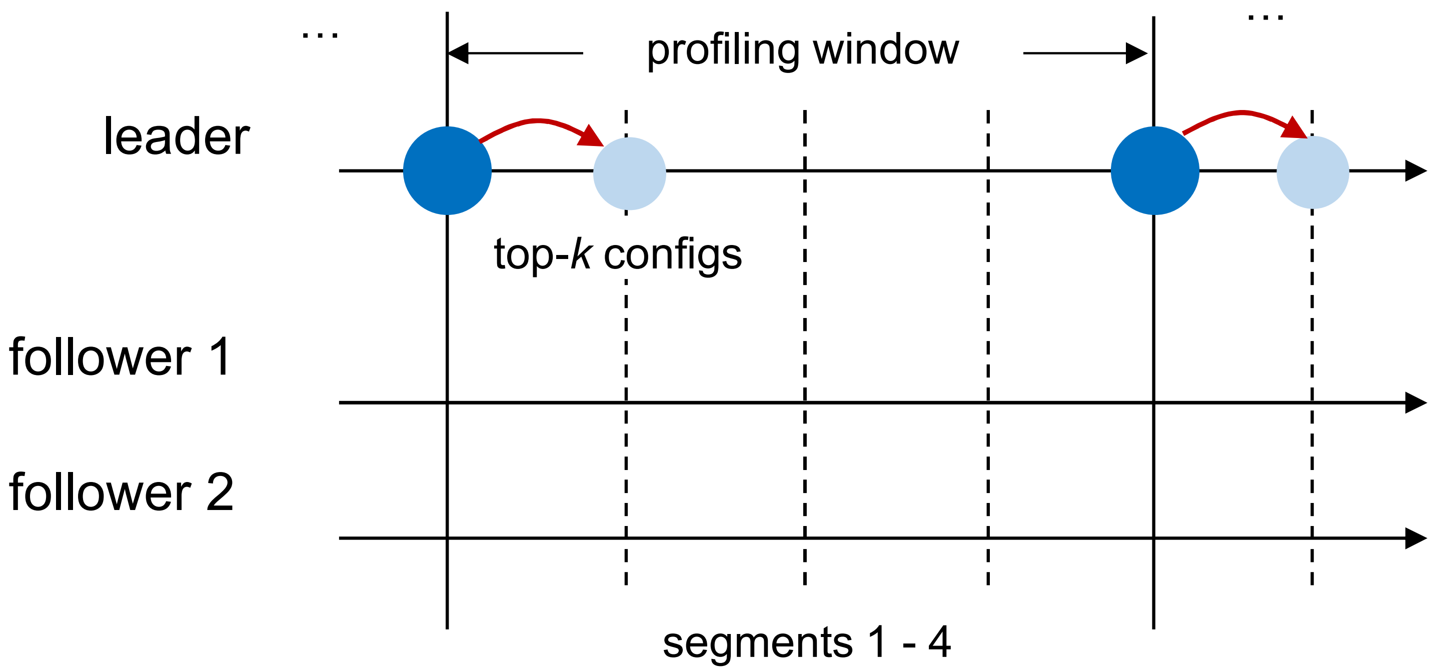
Putting them together (Chameleon)



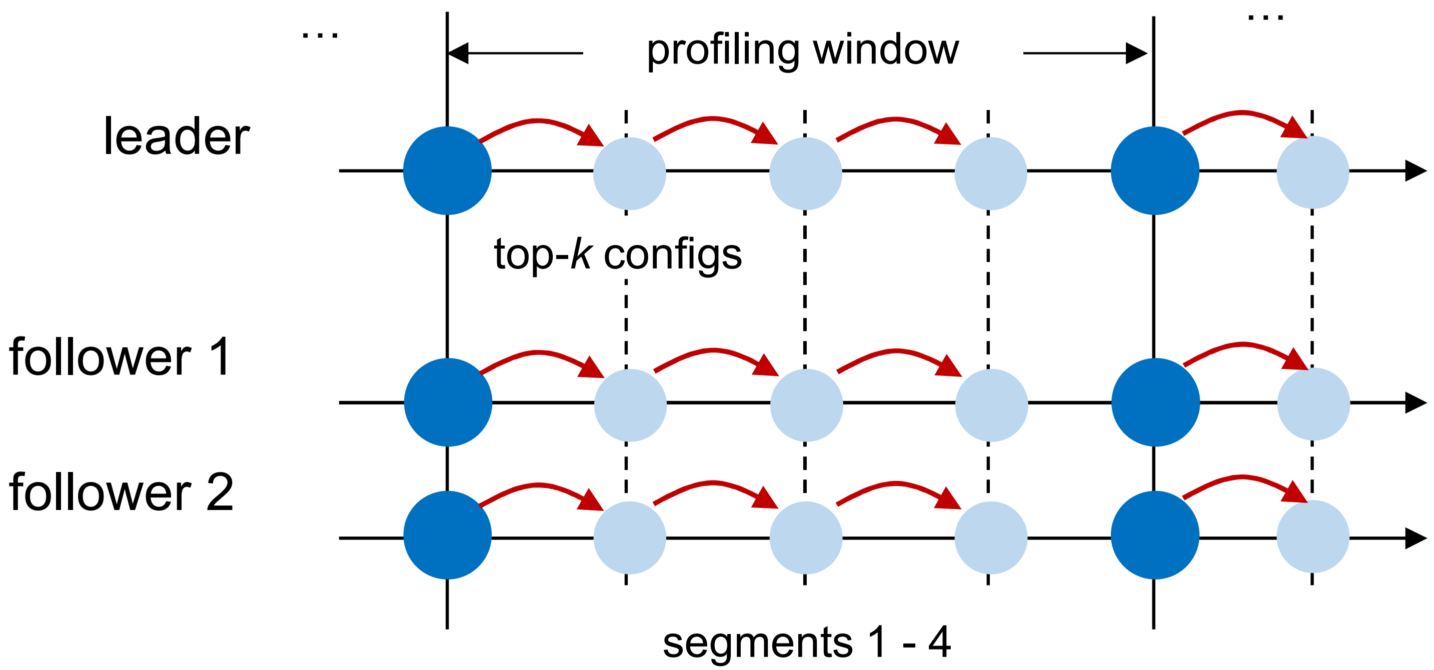
Putting them together (Chameleon)



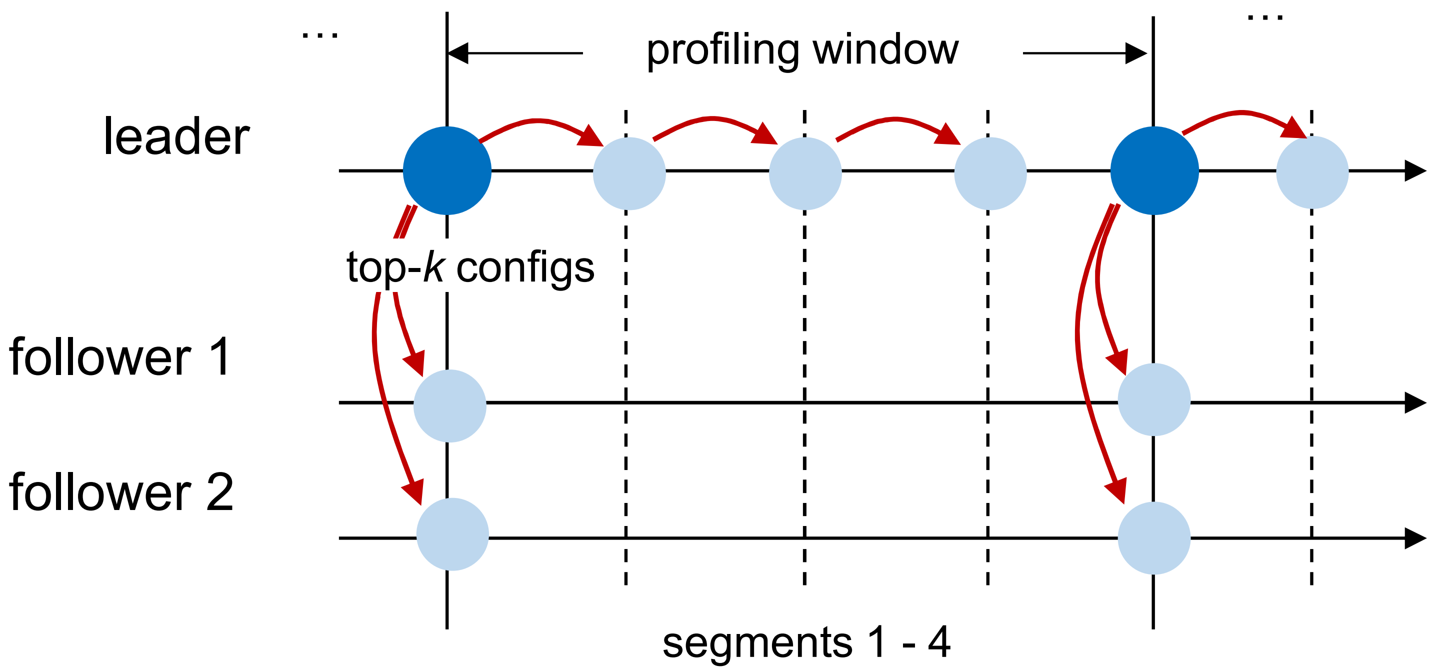
Putting them together (Chameleon)



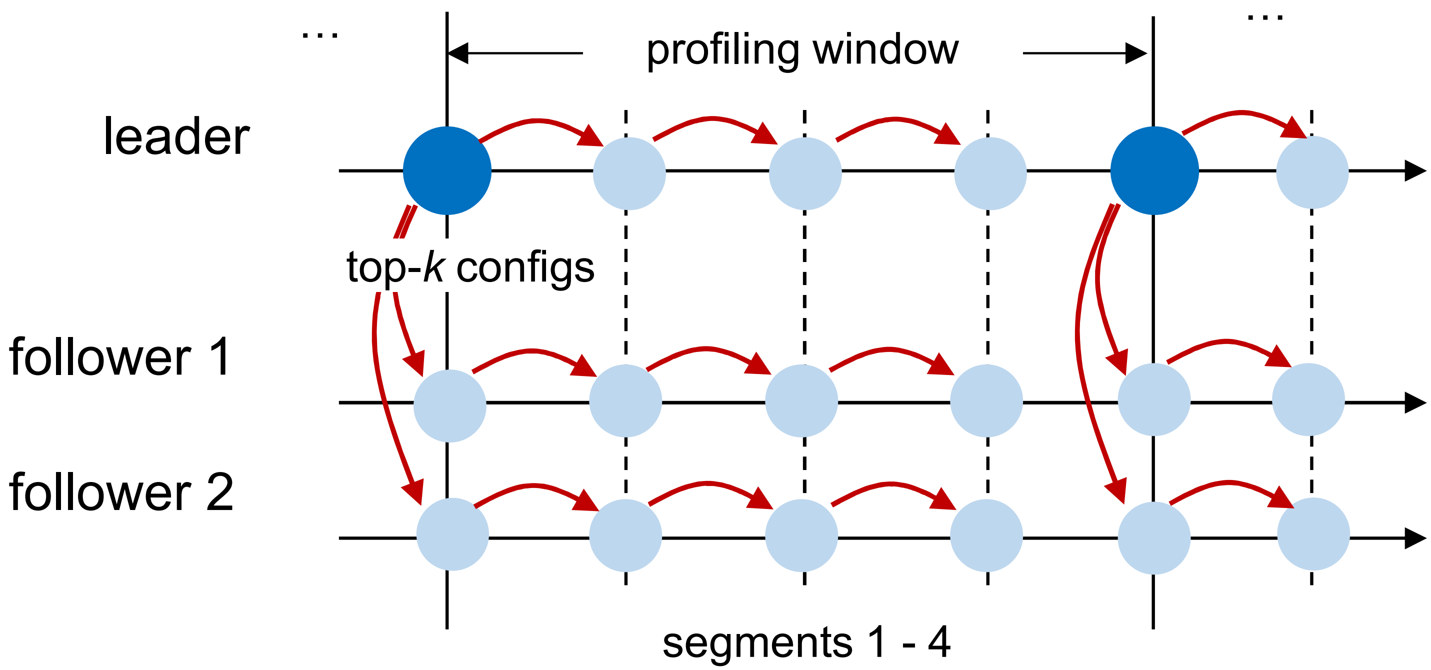
Putting them together (Chameleon)



Putting them together (Chameleon)



Putting them together (Chameleon)

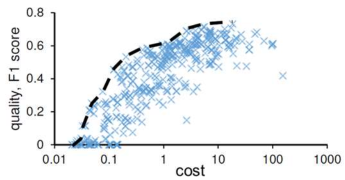


Evaluation Highlights

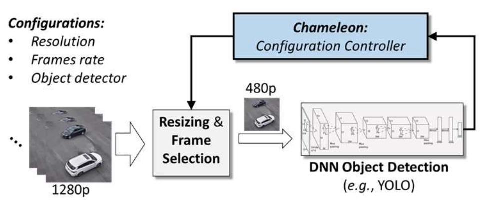
- 2 datasets
  - 5 traffic video cameras at different intersections in Bellevue, WA (120 video clips across 24 hours)
  - 10 cameras in indoor cafeteria (90 video clips across 3 days)
- Chameleon improves accuracy + cost
  - 20%-50% higher accuracy at same cost
  - Same accuracy at 30%-50% of the cost (2-3× speedup)

This talk will cover…

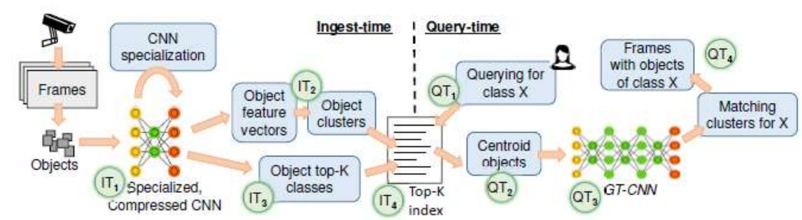
✓ Video analytics pipelines across edge/cloud with *approximation*

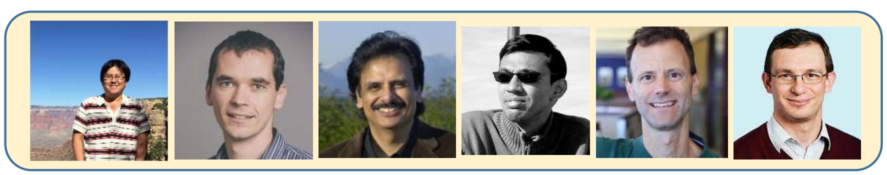


✓ Adaptive video analytics at scale



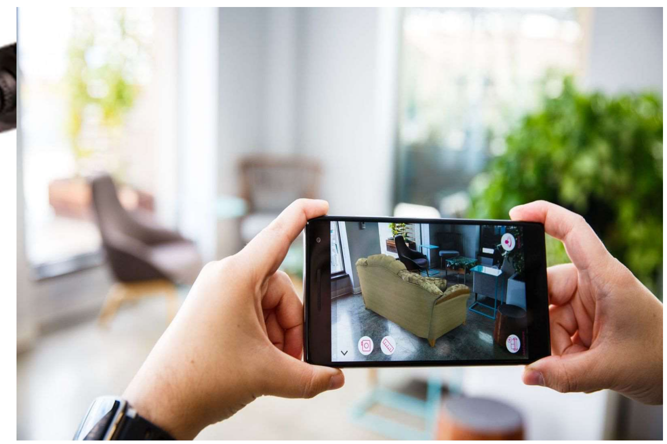
- Interactive querying of stored video datasets





Video recordings are ubiquitous

Massive amounts of video recordings everywhere



Querying on recorded videos is challenging

Convolution Neural Networks (CNNs) enable accurate querying

- *Find all trucks in Bellevue traffic videos yesterday*

slow and costly!

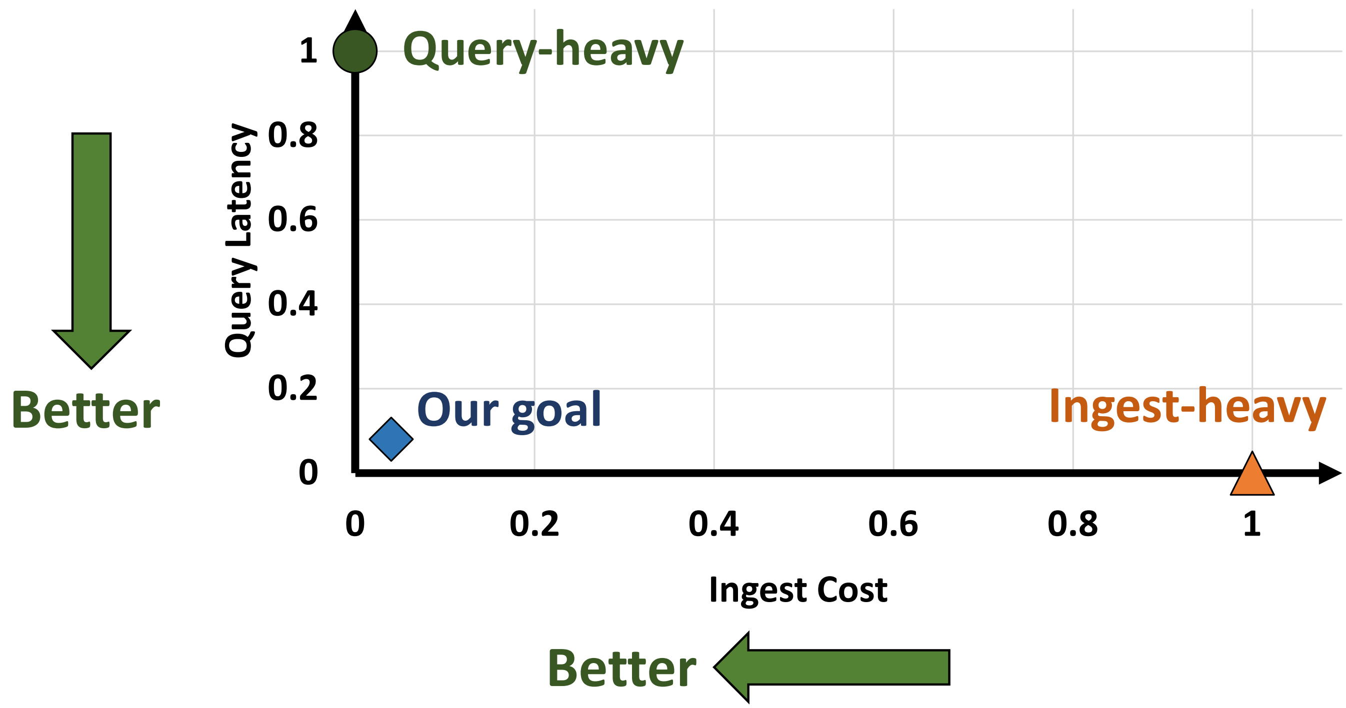
Ingest Time Analysis

- Analyzing all videos at ingest time can make query fast
  - But it is costly and potentially wasteful ($380/month/stream)

Query Time Analysis

- Analyzing videos at query time can save cost
  - But it very slow (5 hr for a month-long video [NoScope @ PVLDB'17])

Enable low-latency, low-cost, and high-accuracy querying over large historical video datasets

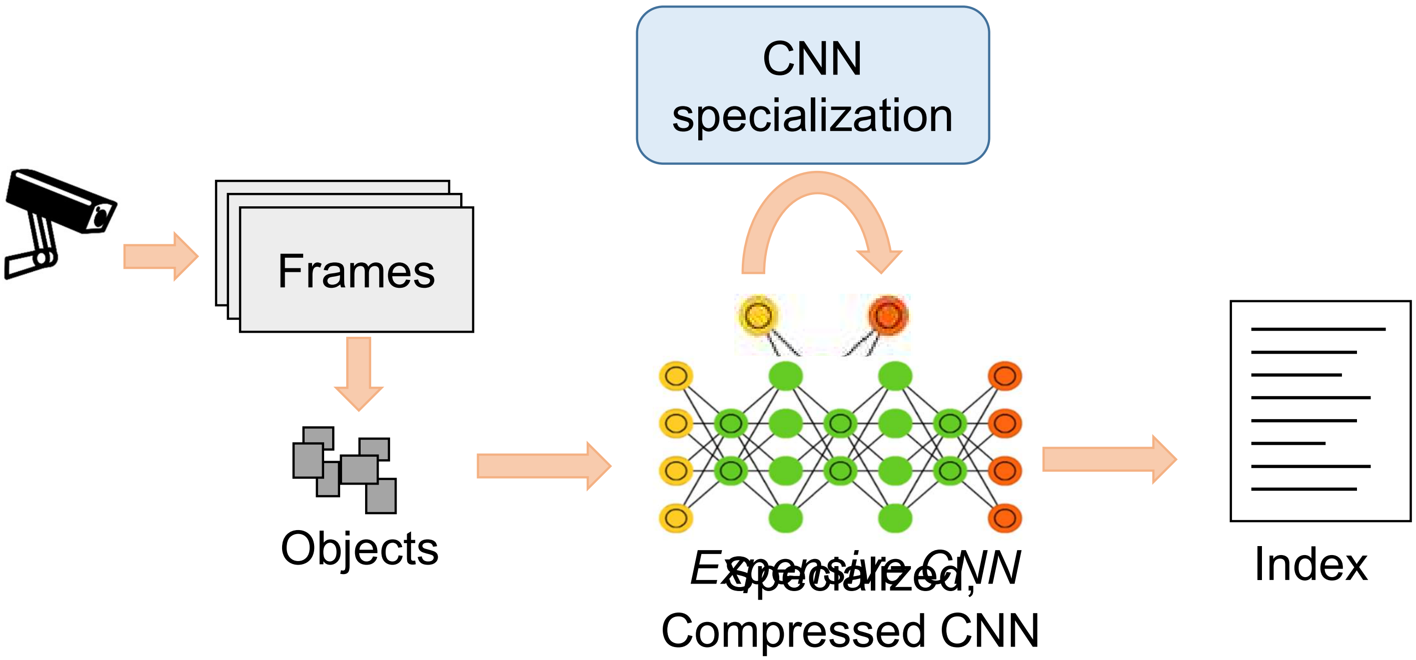


System Objectives

- Provide low-cost indexing at ingest time
- Achieve high accuracy and low latency at query time
- Enable trade-offs between ingest cost and query latency

Low-Cost Ingestion: Cheaper CNNs

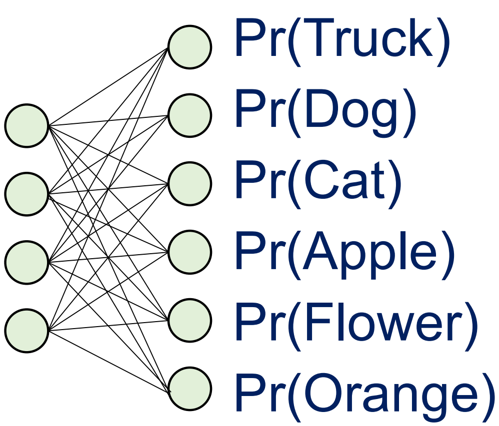
- Process video frames with a cheap CNN at ingest time
  - Compressed and Specialized CNN: fewer layers / weights, and they are specialized for each video stream



Challenge: Cheap CNNs are Less Accurate

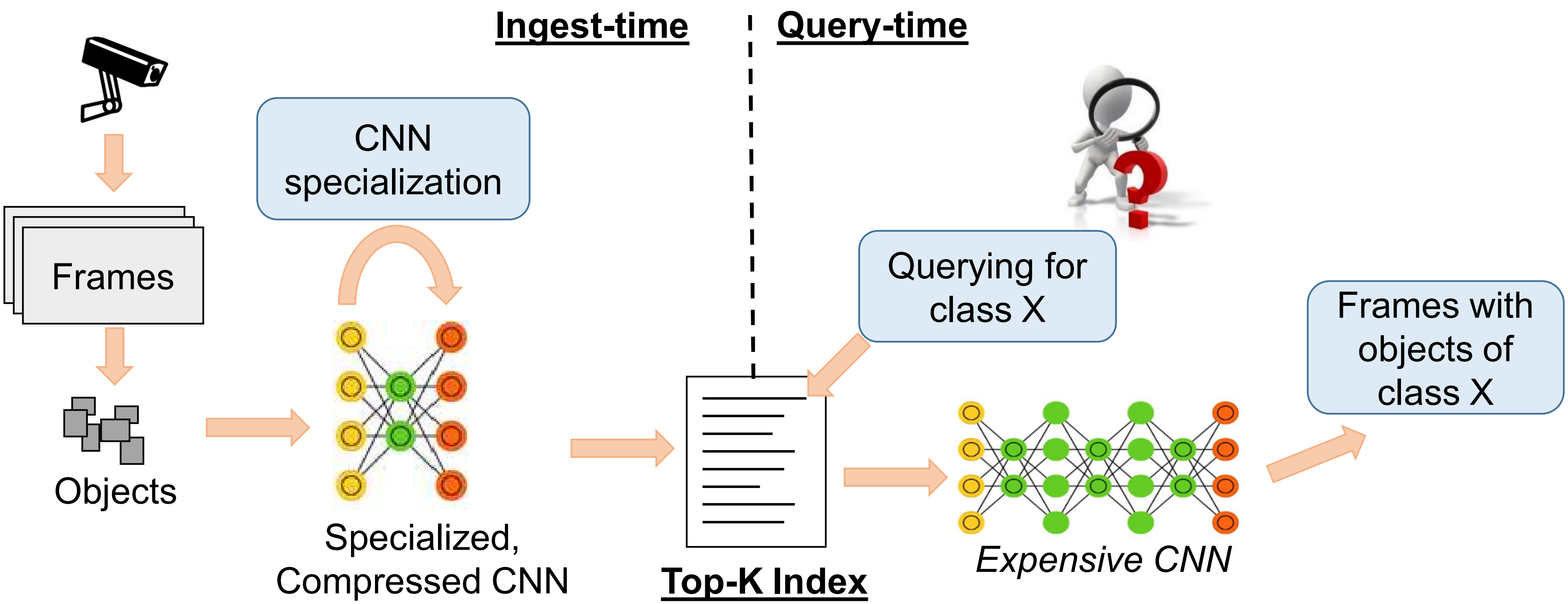
- Cheaper CNNs are less accurate than the expensive CNNs

The best result from the expensive CNN is within the top-K results of the cheaper CNN



Rank	Expensive CNN	Cheap CNN
1	Truck	Moving Van
2	Moving Van	Airplane
3	Passenger Car	Truck
4	Recreational vehicle	Passenger Car

Solution: Top-K Approximate Index

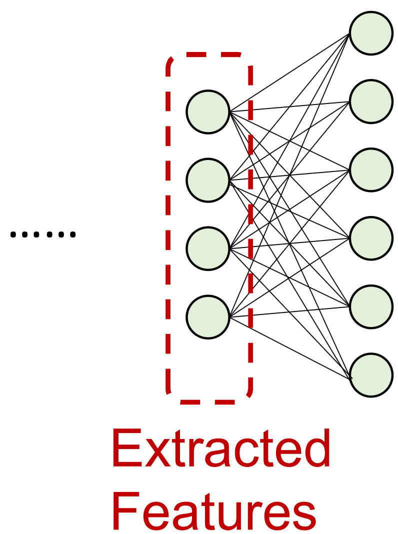


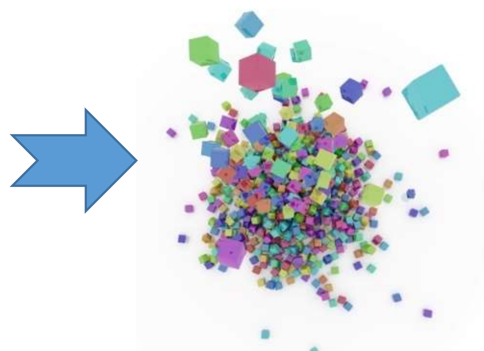
System Objectives

- Provide low-cost indexing at ingest time
- Achieve high accuracy and low latency at query time
- Enable trade-offs between ingest cost and query latency

Low-Latency Query: Redundancy Elimination

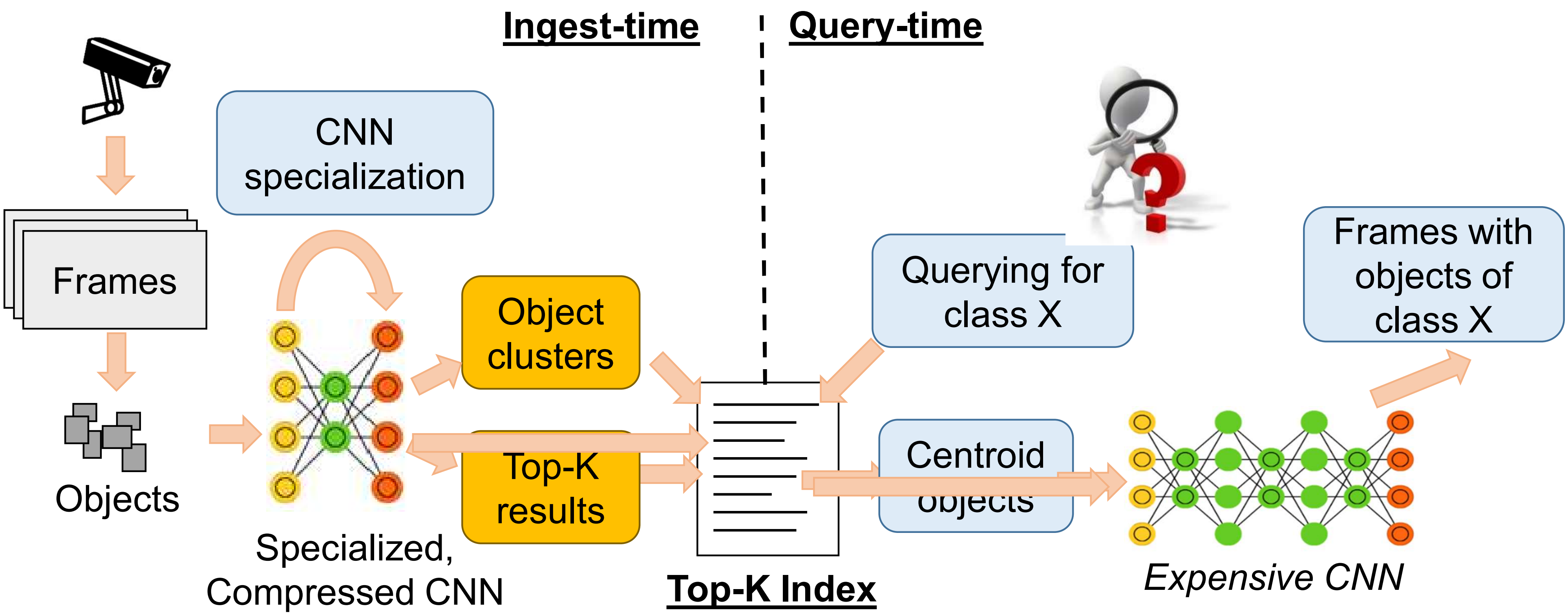
- Approximate indexing ➔ non-trivial work at query time
- Minimize the work at query time ➔ clustering similar objects based on the extracted features
    - Images with similar feature vectors are visually similar [1, 2, 3]





1. Krizhevsky et al., NIPS'12
2. Babenko et al., ECCV'14
3. Razavian et al., CVPR Workshop'14

Adding Feature-based Clustering



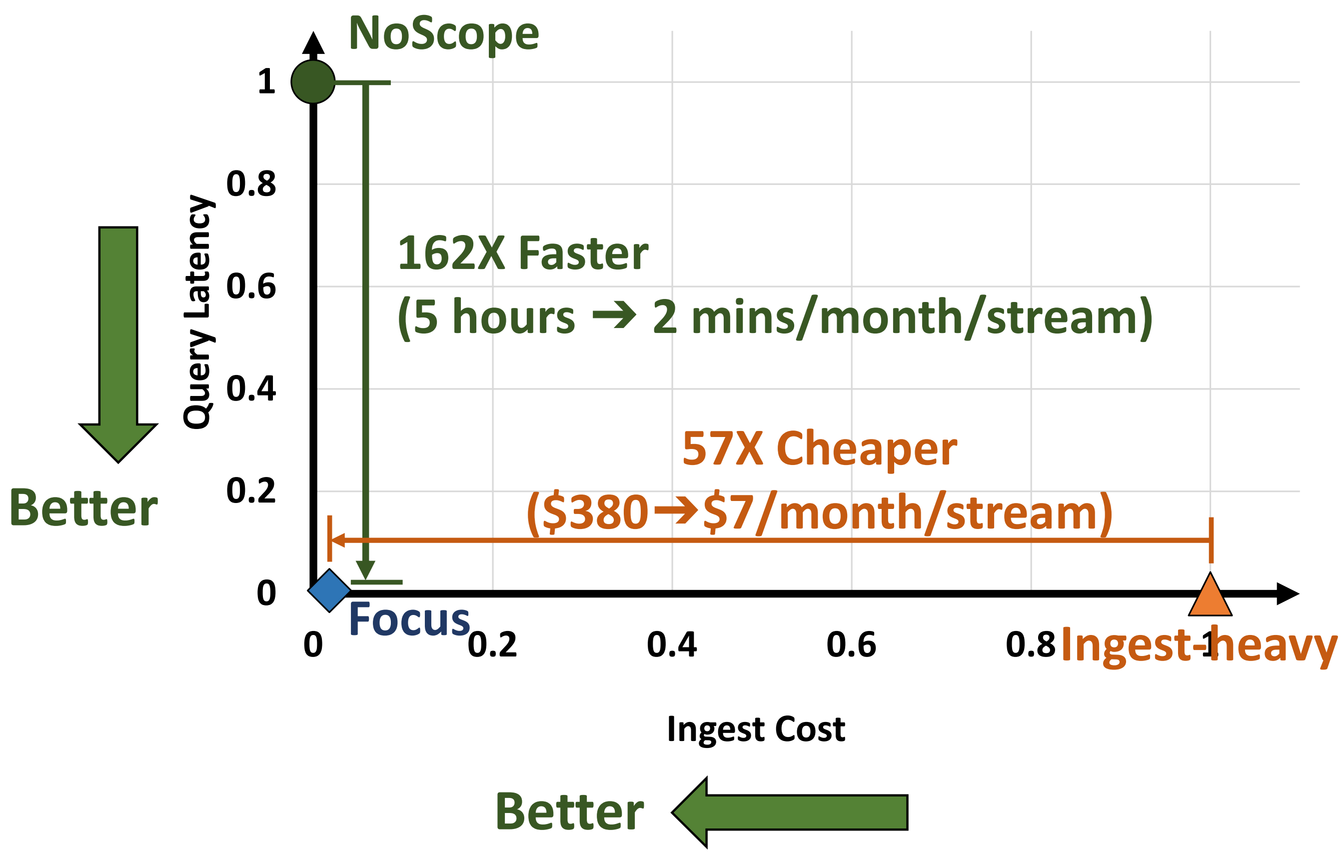
Results Highlights

Video Datasets

Traffic & surveillance videos

Accuracy Targets

Recall & precision – 99%
(w.r.t. YOLOv2)



Focus Demo

Target Recall & Precision of 99%

Baseline is NoScope @ PVLDB'17

- ✓ Frame sampling
- ✓ Binary classifiers for filtering
- ✓ Motion detection

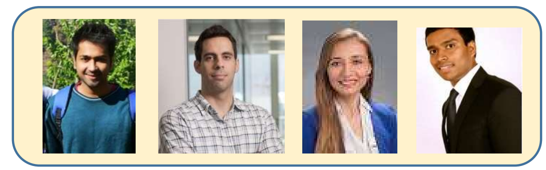
Narrated by Kevin

Ongoing work (*that I did not talk about*)

❑Cross-camera video analytics

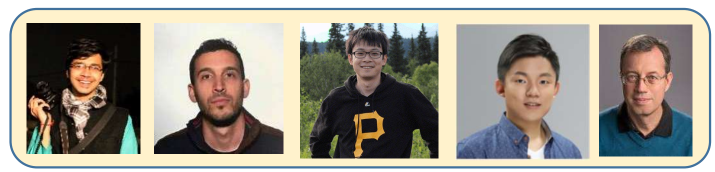
- Large camera deployments in buildings, cities
- Spatio-temporal correlations for efficiency & accuracy

❑Private video analytics as a cloud service



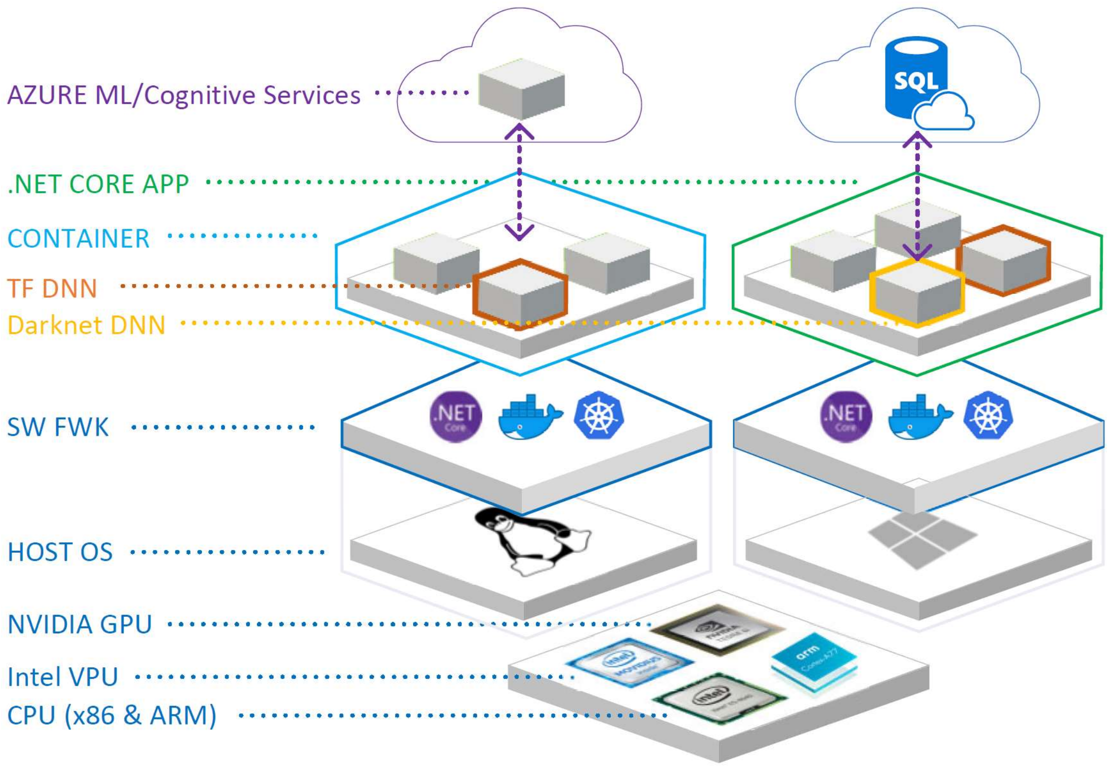
- Side-channel attacks lead to video content leaking
- Hybrid TEE (CPU + GPU enclaves) design for data-obliviousness

❑Continuous model training on edge devices



- Models need to be updated with new data
- Co-existence of training with inference on edge devices

Microsoft Rocket Video Analytics Platform



- Built on C# .NET Core
- Docker containerization
- TensorFlow, ONNX, OpenVINO models
- OpenCV components
- GPU/VPU/FPGA acceleration

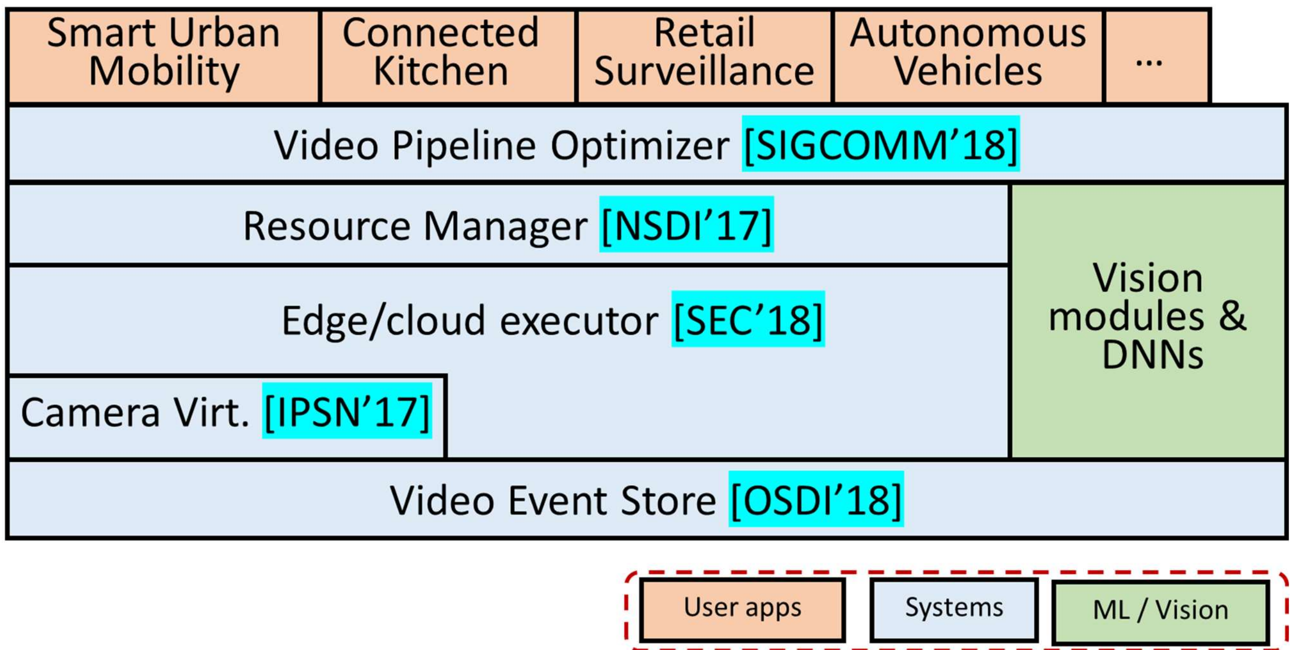
★Code released at https://aka.ms/rocket-oss

Democratizing Video Analytics

✓ Video analytics across edge/cloud with *approximation*

✓ Adaptive video analytics at scale

✓ Interactive querying of stored video datasets



http://aka.ms/rocket

http://aka.ms/ganesh