🙈 NoScope: 🙈

Optimizing Neural Network Queries over Video at Scale

Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, Matei Zaharia

DAWN Project, Stanford InfoLab

http://dawn.cs.stanford.edu/

30 August 2017 @ VLDB 2017

STANFORD INFOLAB

Video is a rapidly growing source of data

- London alone has 500K CCTVs
- 300 hours of video are uploaded to YouTube every minute
- High quality image sensors are incredibly cheap (<$0.70)

We can query video to understand the world

e.g., traffic analysis, environmental monitoring, surveillance, customer behavior, urban dynamics, social science and media studies

running example: when did buses pass by this intersection today?

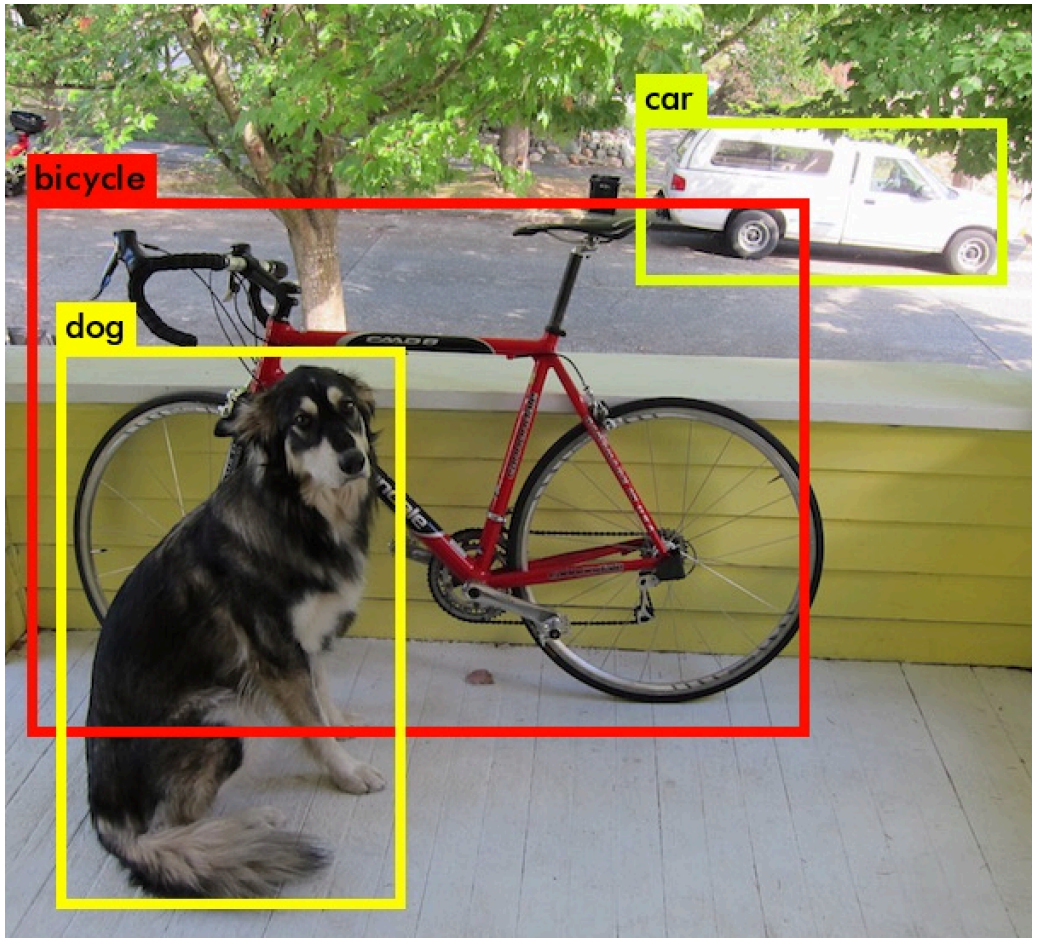
increasingly cheap to acquire this data…

…how to process it?

Computer vision lets us query video automatically

Core capability: Object detection

Input: visual data (e.g., images)



Output: objects and boxes in scene

Neural networks dominate object detection

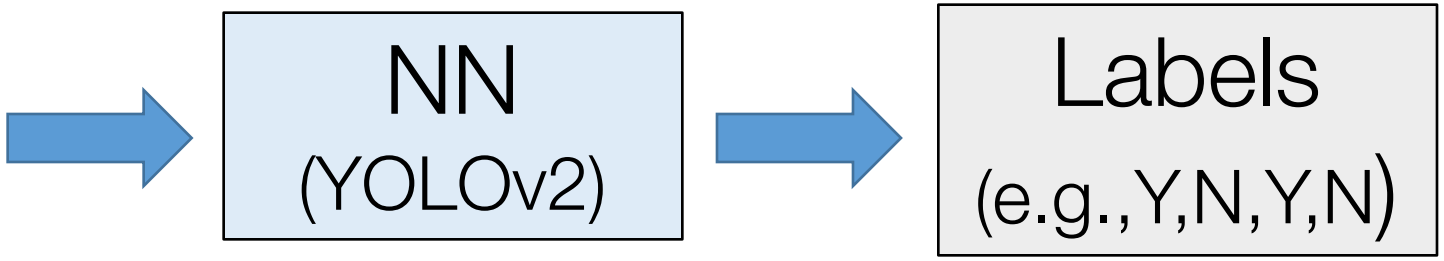
- Idea: many parameters + nonlinear functions capture representations
- Preferred for image analytics, often better than humans
- High-quality models widely available (e.g., open source on GitHub)

Enables new kinds of downstream analytics (e.g., use with DBMS)

Problem: Analysis with NNs doesn't scale

Object detection neural networks evaluate video *one frame at a time*

Video



500K video feeds?

$1B+ of GPUs

NN Hardware	Cost to Purchase	Frames / Second
K80 GPU	$4000	50
P100 GPU	$6000	80

Our research:

Can we make analytics on video scale?

This talk:

NoScope

a system for accelerating neural network video analysis using model specialization and database-inspired query optimization

Outline

- Motivation: Exploding video data demands scalable processing
- NoScope Architecture + Key Contributions
  - Specialized models to exploit query-specific locality
  - Difference detectors to exploit temporal locality
  - Cost-based optimizer for data-dependent model cascade
- Experimental evaluation

NoScope Architecture, Interfaces

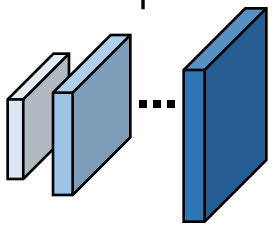
Input:

a) target object

+

b) target video (fixed angle only)

+



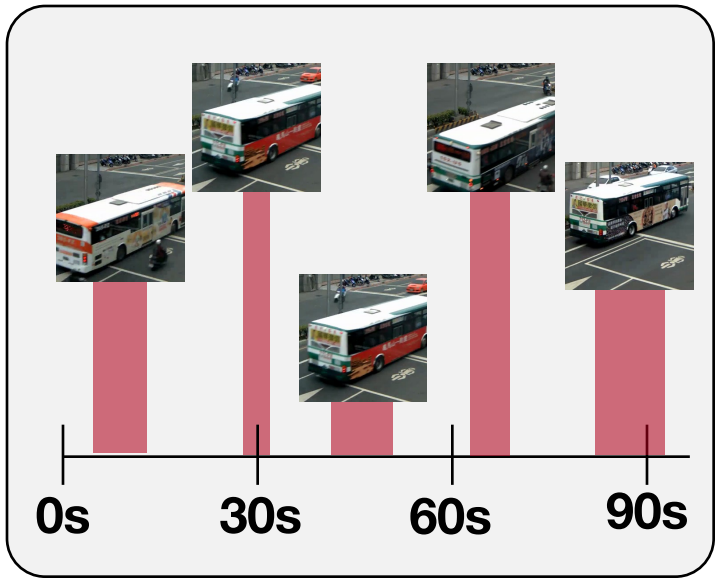
c) reference NN

e.g., "find buses in this webcam feed using YOLOv2"

NoScope

Objective: minimize runtime while mimicking reference NN within target accuracy (e.g., 1%)

Output:



Binary labels over time

e.g., buses appeared at 5-14s, 28-33s, …

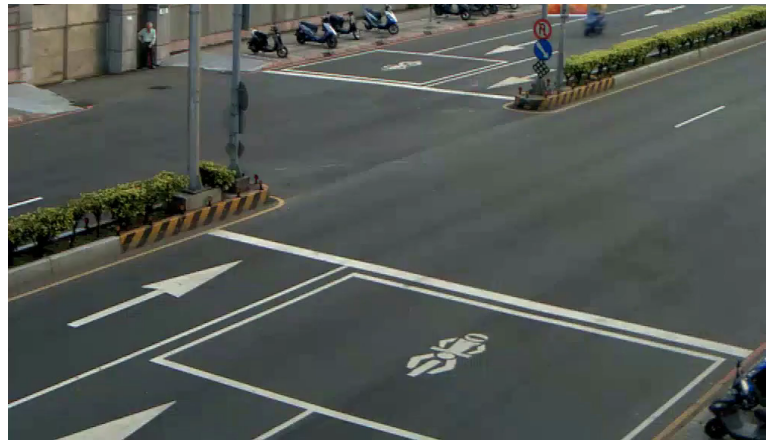
Outline

- Motivation: Exploding video data demands scalable processing
- NoScope Architecture + Key Contributions
  - Specialized models to exploit query-specific locality
  - Difference detectors to exploit temporal locality
  - Cost-based optimizer for data-dependent model cascade
- Experimental evaluation

Opportunity 1: Query-specific locality

Query: "When did buses pass by this intersection?"





Target objects appear from similar perspectives in video

Opportunity 1: Query-specific locality

Query: "When did buses pass by this intersection?"

NNs are typically trained to detect
tens of object categories
in **arbitrary scenes**
and from **arbitrary angles**

If we only want to detect
buses in a given video,
we're overpaying

images from
training set for YOLOv2

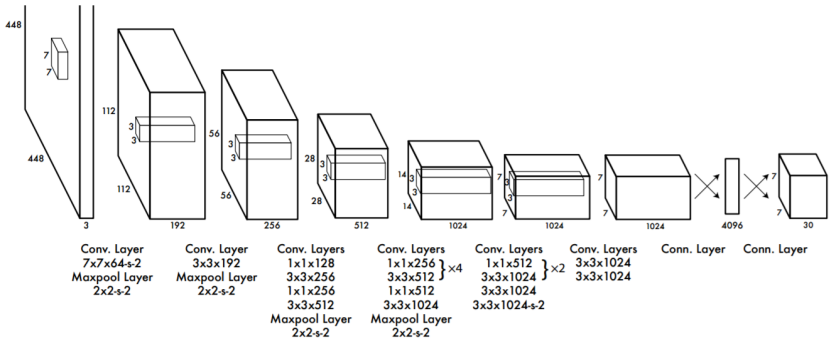
Key idea: specialize for query and video

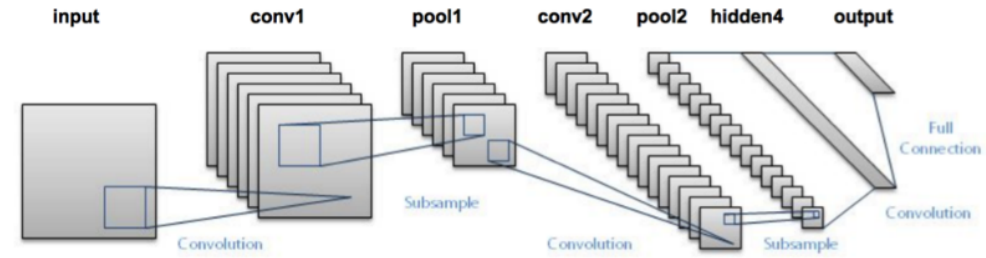
Idea: use big reference NN to train a smaller, specialized NN

The specialized NN:

Only works for a given video feed and object

Is much, much smaller than the reference NN





Specialized models are much smaller

YOLOv2

24 convolutional layers

64-1024 filters per layer

4096 neurons in FC layer

35 billion FLOPS

NoScope specialized model

4 convolutional layers

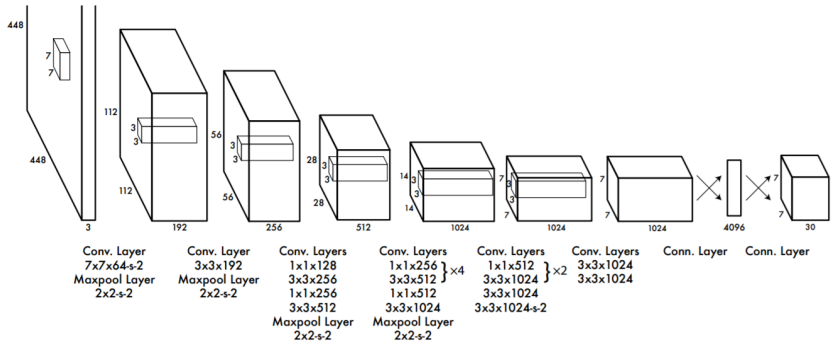
32-128 filters per layer

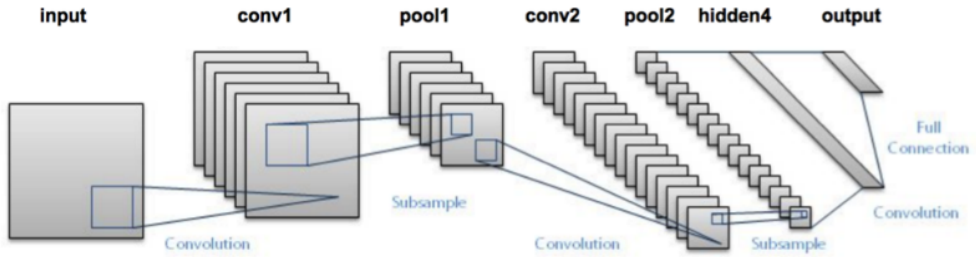
32 neurons in FC layer

3 million FLOPS

10,000x fewer FLOPS

Specialized models are much faster





YOLOv2:

80 fps

Specialized NN:

25k+ fps

300x faster execution on GPU

Specialization != Model Compression

Model compression/distillation [NIPS14, ICLR16]: lossless models

Goal: smaller model for **same task** as reference model

Result: typically **2-10x** faster execution

Specialization: perform "lossy" compression of reference model

A specialized model **does not generalize** to other videos…

…but is accurate on target video, up to **300x** faster

NoScope's Model Specialization Procedure

1. **Run big NN** for few hours to obtain video-specific training data
2. **Train specialized NN** over video-specific training data
3. **Enable specialized NN**, only call big NN when unsure

In paper: NoScope automatically searches for the smallest NN

Outline

- Motivation: Exploding video data demands scalable processing
- NoScope Architecture + Key Contributions
  - Specialized models to exploit query-specific locality
  - Difference detectors to exploit temporal locality
  - Cost-based optimizer for data-dependent model cascade
- Experimental evaluation

Opportunity 2: Temporal locality

Query: "When did buses pass by this intersection?"

Both videos run at 30 frames per second,
requiring 30 NN evaluations per second

Opportunity 2: Temporal locality

Query: "When did buses pass by this intersection?"

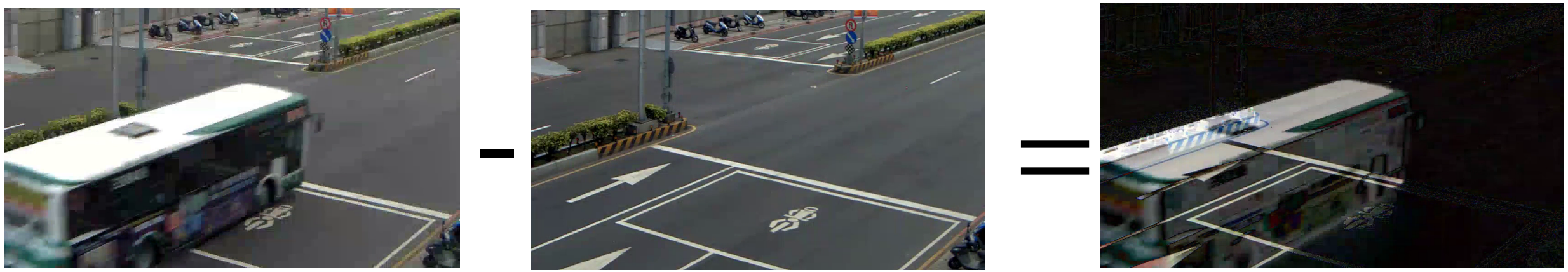
Observation: frames close in time are often redundant

NoScope: train a fast model to detect redundancy

Difference detection: detect redundant frames

Many techniques in the literature for detecting scene changes

NoScope: simple regression model over subtracted frames



Frame 0 | Frame 1 | Difference

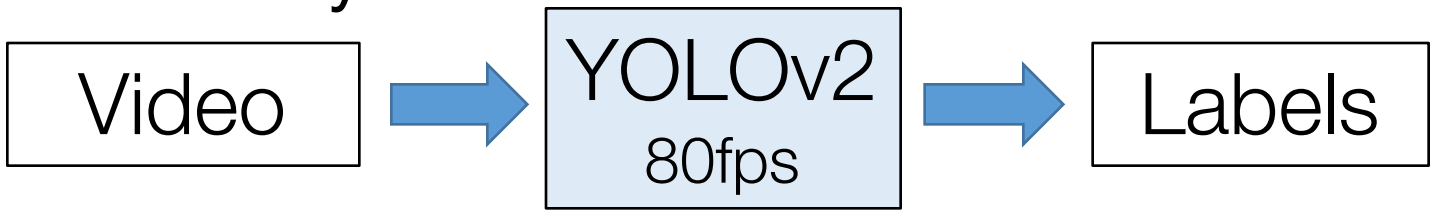
Surprising: detecting differences is faster than even specialized NNs

Difference detection runs at 100k+ fps on CPU

Outline

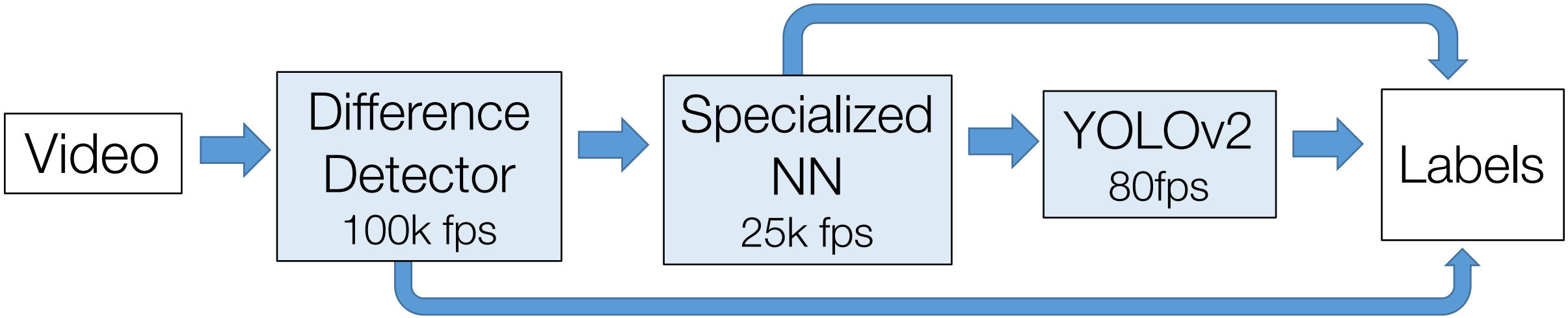
- Motivation: Exploding video data demands scalable processing
- NoScope Architecture + Key Contributions
  - Specialized models to exploit query-specific locality
  - Difference detectors to exploit temporal locality
  - Cost-based optimizer for data-dependent model cascade
- Experimental evaluation

Previously:



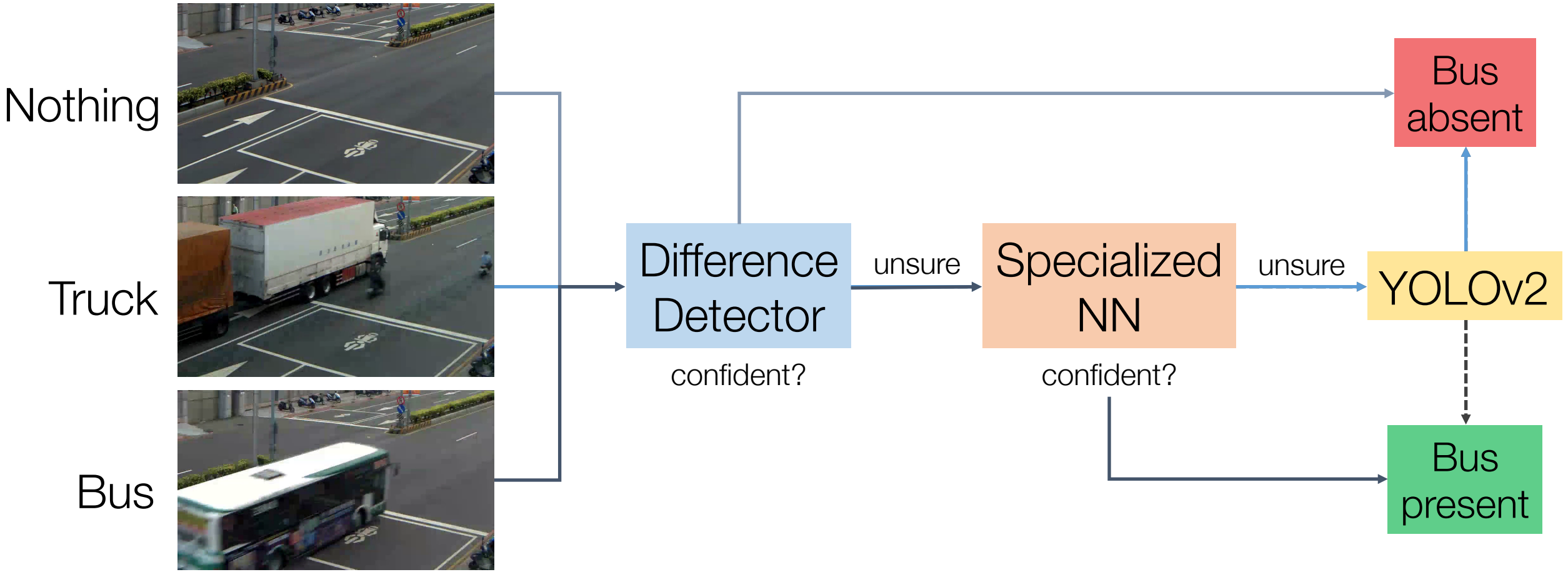
NoScope combines fast models in a cascade

[Viola, Jones CVPR 2001]



Idea: Use the cheapest model possible for each frame

Cascades avoid unnecessary computation on each frame



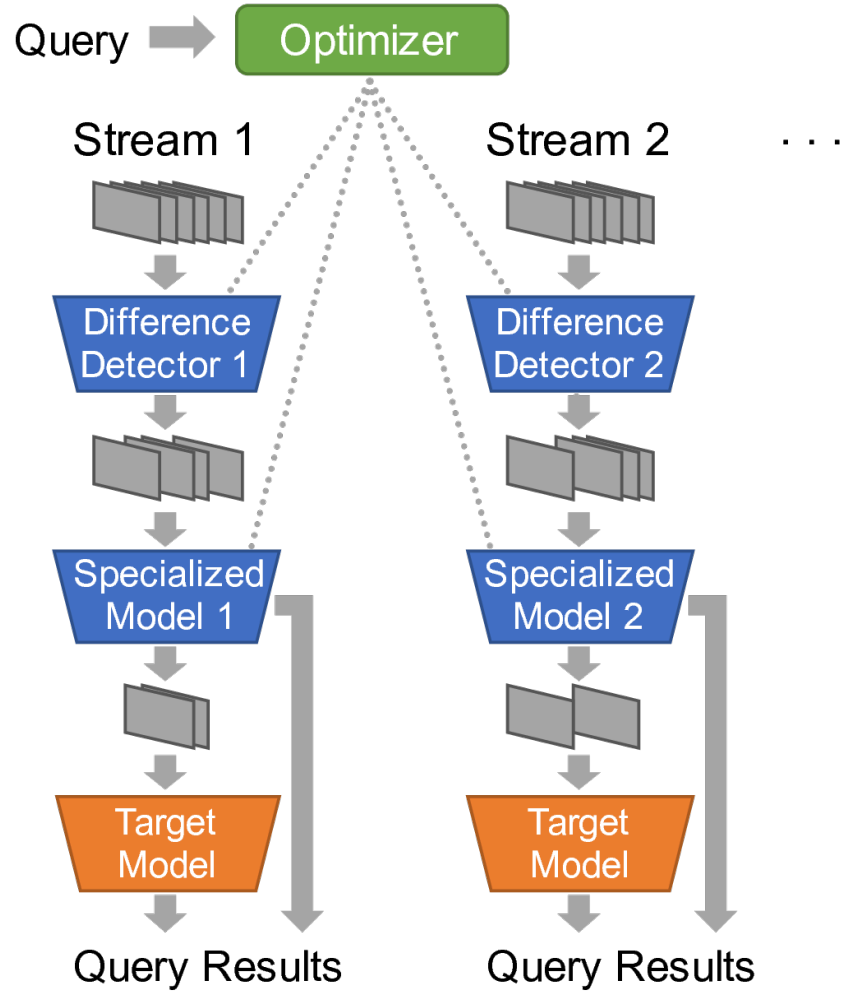
NoScope performs cost-based optimization for cascades

Given an accuracy target, NoScope performs:

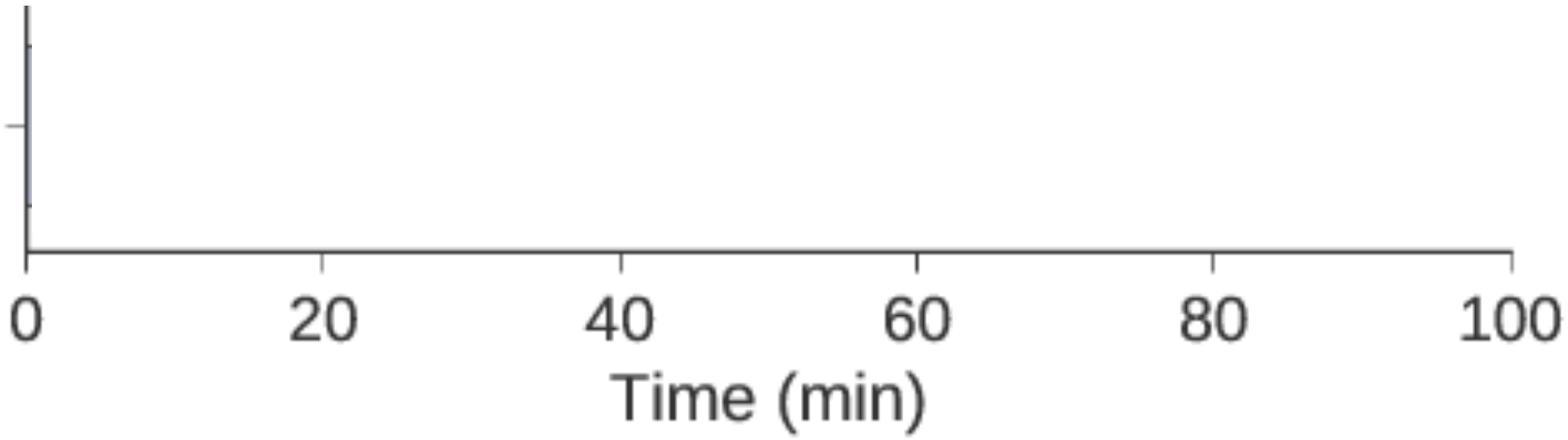
Model search: e.g., how many layers in specialized NN?

Cascade search: e.g., how to set the cascade thresholds?

Data-dependent process: high-quality choices vary across queries and videos (see paper)



Typical NoScope Query Lifecycle



Current Limitations (cf. Section 8)

Targets **binary detection** tasks (e.g., bus/no bus)

- Ongoing research on also locating objects (e.g., bus location)

Targets **fixed-angle cameras** (e.g., surveillance cameras)

- Ongoing research on moving cameras

Does not automatically **handle model drift**

- Requires representative training set (e.g., morning, afternoon)

Batch-oriented processing

- Poor on-GPU support for control flow in cascades

Outline

- Motivation: Exploding video data demands scalable processing
- NoScope Architecture + Key Contributions
  - Specialized models to exploit query-specific locality
  - Difference detectors to exploit temporal locality
  - Cost-based optimizer for data-dependent model cascade
- Experimental evaluation

Experimental configuration and videos

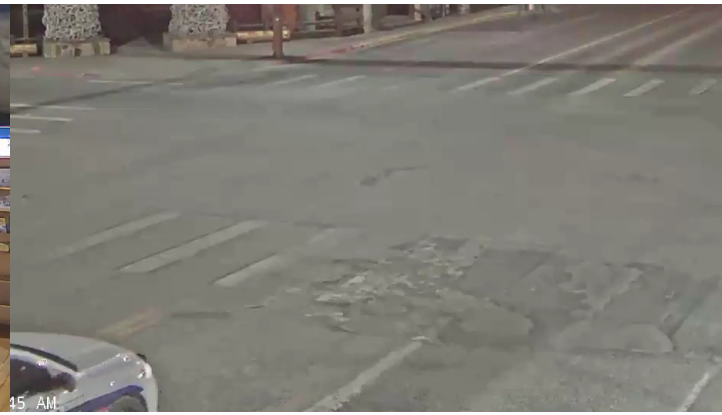
System setup: Difference detectors run on 32 CPU cores + specialized/target NNs runs on P100 GPU; omit MPEG decode time

Seven video streams from real-world, fixed-angle surveillance cameras; 8-12 hours of video per stream (evaluation set)

Taipei: bus

Amsterdam: car

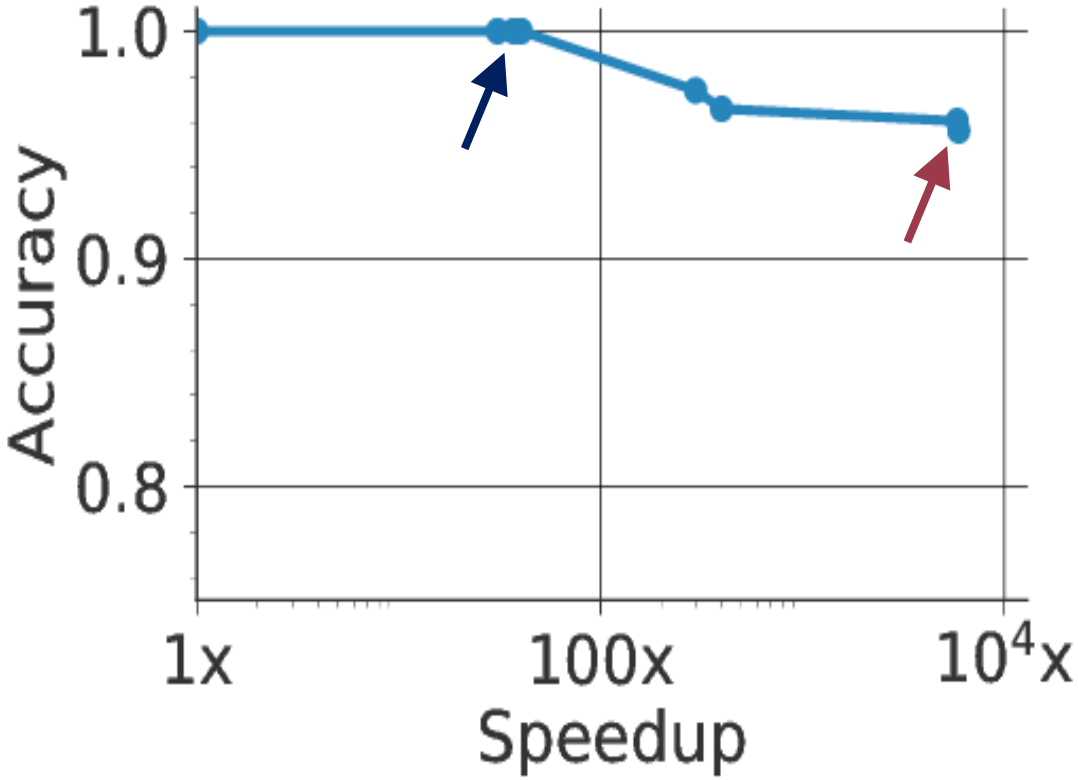
Store: person



Jackson Hole: car

NoScope enables accuracy-speed trade-offs

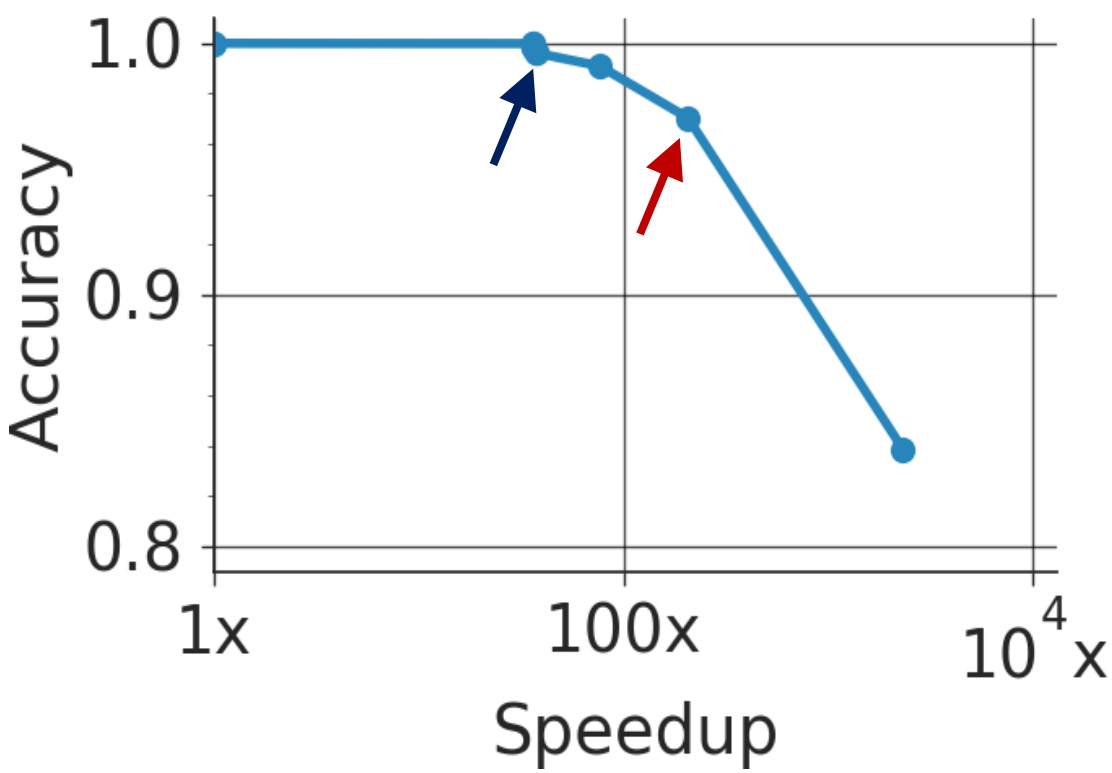
Elevator (best result)



40x faster @ 99.9% accuracy

5858x faster @ 96% accuracy

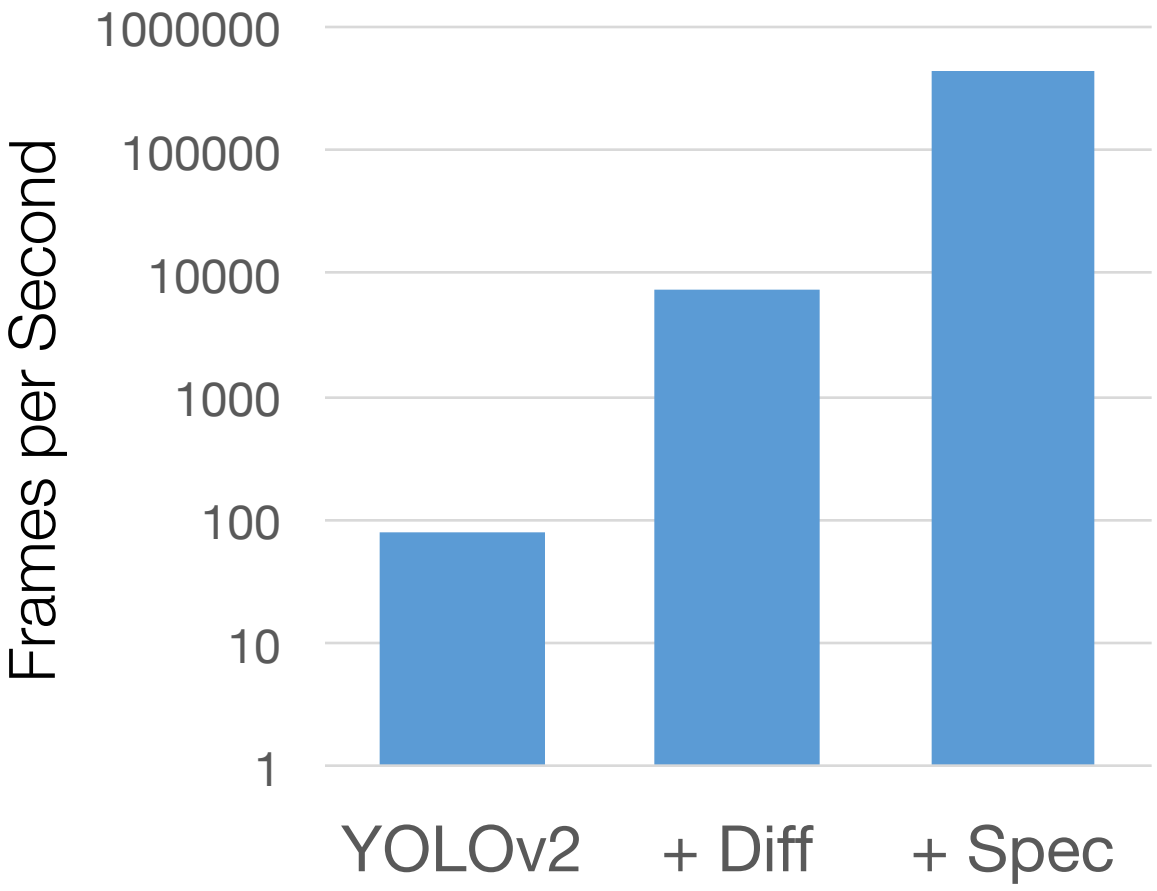
Taipei (worst result)



36.5x faster @ 99.9% accuracy

206x faster @ 96% accuracy

Factor Analysis: All components contribute to speedups



video: elevator **false positives:** 1% **false negatives:** 1%

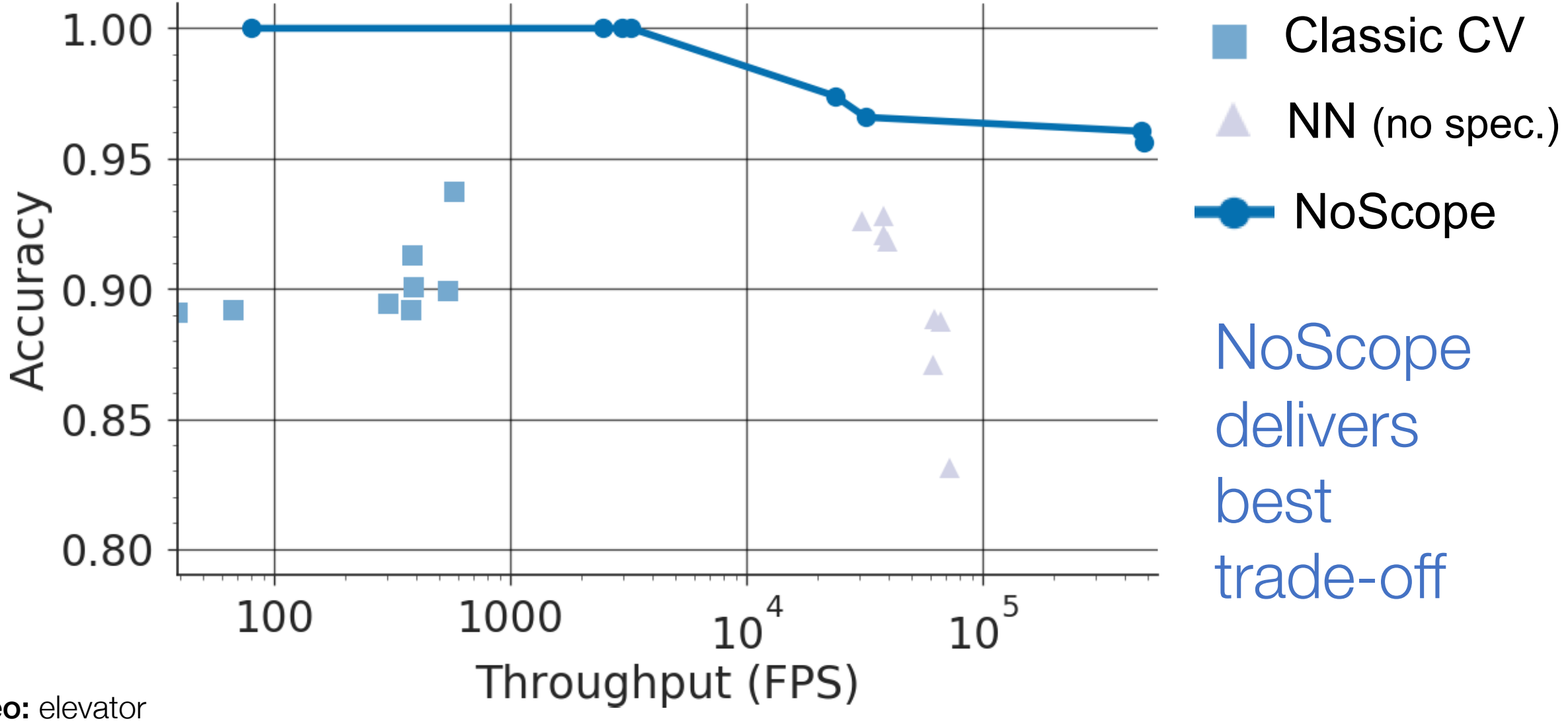
For this video:

Difference detection can filter 95% of frames

Specialized models can filter *all* remaining frames

Similar trends for other videos, depending on content

Comparison w/ classic methods, non-specialized NNs



NoScope delivers best trade-off

video: elevator

Additional content in paper

» Lesion study evaluating contribution of each optimization

» Demonstration of optimizer selection procedure

» Efficient firing threshold search for optimizer

» Additional details on limitations and extensions

» Additional related work for computer vision, NNs, RDBMS

Conclusions

Neural networks can automatically analyze rapidly growing video datasets, but are very slow to execute (50-80fps on GPU)

NoScope accelerates NN-based video queries by:

1. Specializing networks to exploit query-specific locality
2. Training difference detectors to exploit temporal locality
3. Cost-based optimization for video-specific cascades

Promising results (10-1000x speedups) for many queries

`https://github.com/stanford-futuredata/noscope`