Reinventing Video Streaming for Distributed Vision Analytics

Chrisma Pakha, Aakanksha Chowdhery, Junchen Jiang
Automated video analytics everywhere

Deep Neural Networks

“find speeding cars on highway”
“detect an object of interest”
High accuracy at the expense of higher cost

Deep Neural Networks

- Computationally expensive
DNN + Video can be prohibitively expensive at scale
Processing one live feed can be costly

One camera of **30 fps**

**Compute**
($9,999$ GPU or $$$$/mo cloud)

**Network**

**Storage**
What if we have 1000x more cameras?
New York City has 4,176 cameras below 14\textsuperscript{th} Street...

What if videos are 3840 × 2160 (4K)?
Most cameras today are only 1280 x 720

How to achieve higher accuracy at lower cost/bandwidth use?
Custom Video Streaming stack to explicitly balance bandwidth-accuracy tradeoffs
Video Streaming Stack: Bandwidth vs. Accuracy

Client (Camera) → Video Streaming Stack → Server

Need Custom Video Streaming: higher accuracy at lower cost/bandwidth use?
MPEG for Distributed Vision Analytics

Client (Camera)

Video Streaming Stack

Server

MPEG video encoding

Maximizes user-perceived QoE

• Max resolution
• Min video re-buffering/stalling
• Min frames dropped

Agnostic to DNN-perceived QoE
Client-side filtering for Distributed Vision Analytics

- **Client (Camera)**
- **Server**
  - **Video Streaming Stack**
  - **Deep Neural Network (DNN)**

**Client-side filtering**
- Less accurate, may miss critical details

- Heuristics: frame difference detection (Glimpse, NoScope)
- Less accurate vision models (Vigil)
- Specialized DNNs (MCDNN, NoScope)

**Limited computational capability**
State-of-the art approaches

**Video Streaming Stack**

- **Client (Camera)**
- **Server**

**Baseline #1: MPEG video encoding**
- Maximizes user-perceived QoE

**Baseline #2: Client-side filtering**
- Less accurate, may miss critical details

Both are agnostic to the DNN logic

⇒ Suboptimal bandwidth-accuracy tradeoffs
Design of DNN-driven Streaming

- Fetch video segments of interest

  ➔ Inference accuracy at **lower resolution**: *likely objects*

![Whole image at low resolution](image1)

- **umbrella** – 41.4%
- **truck** – 60.7%
- **car** – 61.3%

![Cropped areas at high resolution](image2)

- **no object**
- **bus** – 67.3%
- **car** – 83.7%
Design of DNN-driven Streaming

- Fetch video segments of interest
  - Inference accuracy at **lower resolution**: likely objects
  - Inference accuracy in **sparsely sampled frames**: likely object locations
  - Focus on **region-of-interest** (Cropping)
DNN-driven Streaming : Optimal

250x bandwidth savings!

Better

Client-side filtering
MPEG encoding
Optimal
DNN-driven Streaming: Iterative Workflow

DNN-driven streaming explicitly balances between accuracy and bandwidth
Preliminary results achieve better bandwidth-accuracy tradeoffs

Accuracy (F1 score) vs Bandwidth Consumption (Kbps)

- Accuracy > 0.95
- 20x bandwidth savings!

Client-side filtering
Preliminary design
MPEG encoding
Optimal

Better
Preliminary results achieve better bandwidth-accuracy tradeoffs

- Better 4-23x bandwidth savings, Accuracy > 0.95

Gains depend on video content

Client-side filtering
MPEG encoding
Preliminary design
Optimal
Final remarks

• Accurate video analytics is increasingly needed!
  However, applying DNN on videos can be prohibitively expensive at scale

• Better bandwidth-accuracy tradeoff by custom video streaming stack
  Key insight: Streaming stack should be driven by the DNN logic

• Promising order-of-magnitude bandwidth savings!
  Several practical challenges remain...