Approximate Query Service on Autonomous IoT Cameras (MobiSys 2020)

Mengwei Xu\textsuperscript{1}, Xiwen Zhang\textsuperscript{2}, Yunxin Liu\textsuperscript{3} Gang Huang\textsuperscript{1}, Xuanzhe Liu\textsuperscript{1}, Felix Xiaozhu Lin\textsuperscript{2}  
\textsuperscript{1}Peking University, \textsuperscript{2}Purdue University, \textsuperscript{3}Microsoft Research

Presenter: Rui Yang  
Date: Oct.22, 2020
Video Analytics Apps

- Busy cross roads
- Retailing store
- Sports stadium
- Parking lots
- ...

Urban, residential areas
- Wired electricity
- Good internet
Video Analytics Apps

- Busy cross roads
- Retailing store
- Sports stadium
- Parking lots
- ...

Urban, residential areas

Rural, off-grid area

- Construction sites
- Cattle farms
- Highways
- Wildlifes
- ...

?
Autonomous Camera

- Energy-independent and Compute-independent

Small-sized energy harvester e.g., “10Wh today”

Commodity SoCs, RPI-like, chargeable battery
Autonomous Camera

- **Energy-independent** and **Compute-independent**

Small-sized energy harvester e.g., “10Wh today”

Commodity SoCs, RPI-like, chargeable battery

Concise, numerical video summaries
Elf for Autonomous Cameras

- Target video query: object counting (with bounded error)

Objective: Min (mean CI width)

[450±40] [600±55] [50±15]

(a) A horizon

(b) A window
Elf for Autonomous Cameras

- Target video query: object counting

Query: (car, 30 mins)

- Install
- Sample & capture

- 7:00AM-7:30AM: [500 ± 100] cars
- 7:30AM-8:00AM: [700 ± 140] cars
- 8:00AM-8:30AM: [800 ± 180] cars
- 8:30AM-9:00AM: [400 ± 100] cars
- 9:30AM-10:00AM: [200 ± 80] cars

95% confidence

-80 200 +80

confidence interval
Elf for Autonomous Cameras

- Target video query: object counting

- The central problem: planning constrained energy for counting
  - Energy model: a budget that cannot be exceeded in a horizon (e.g., 24 hrs)
  - Target: smallest mean CI widths across all (30-min) windows in a horizon
  - Trade-offs: frame sampling and NN selections
Elf for Autonomous Cameras

- Target video query: object counting

- The central problem: planning constrained energy for counting
  - Energy model: a budget that cannot be exceeded in a horizon (e.g., 24 hrs)
  - Target: smallest mean CI widths across all (30-min) windows in a horizon
  - Trade-offs: frame sampling and NN selections

- Solution: two main aspects:
  - Per window: characterizing count actions and outcome
  - Across windows: making joint count decisions on the go
Elf Overview

Energy budget

[450±40] [600±55] [50±15]

Time

Next horizon

Object counts with CIs

Sampled frames

Selected Neural Net

Aggregator with error Integration

Reinforcement Learning Planner

Camera Operating System
Elf tech #1: per-window energy/CI fronts

- What’s the best count action for a window?
  - A count action: determining (1) an NN and (2) # of frames to process

Energy Consumption = $E(\text{NN}) \times \text{frame\textunderscore num}$
Elf tech #1: per-window energy/CI fronts

- What’s the best count action for a window?
  - A count action: determining (1) an NN and (2) # of frames to process

Energy Consumption = E(NN) \times \text{frame\_num}

<table>
<thead>
<tr>
<th>NN Counters</th>
<th>Input</th>
<th>mAP</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv3 (Golden, GT) [85]</td>
<td>608x608</td>
<td>33.0</td>
<td>1.00</td>
</tr>
<tr>
<td>YOLOv2 [84]</td>
<td>416x416</td>
<td>21.6</td>
<td>0.22</td>
</tr>
<tr>
<td>faster rcnn inception-v2 [86]</td>
<td>300x300</td>
<td>28.0</td>
<td>0.40</td>
</tr>
<tr>
<td>ssd inception-v2 [68]</td>
<td>300x300</td>
<td>24.0</td>
<td>0.08</td>
</tr>
<tr>
<td>ssd mobilenet-v2 [88]</td>
<td>300x300</td>
<td>22.0</td>
<td>0.05</td>
</tr>
<tr>
<td>ssdlite mobilenet-v2* [88]</td>
<td>300x300</td>
<td>22.0</td>
<td>0.04</td>
</tr>
</tbody>
</table>
Elf tech #1: per-window energy/CI fronts

- What’s the best count action for a window?
  - A count action: determining (1) an NN and (2) # of frames to process

When energy is low: cheaper NNs win
- Bottlenecked by sampling error (frame quantity)
Elf tech #1: per-window energy/CI fronts

- What’s the best count action for a window?
  - A count action: determining (1) an NN and (2) # of frames to process

When energy is low: cheaper NNs win
  - Bottlenecked by sampling error (frame quantity)

When energy is low: more accurate NNs win
  - Bottlenecked by NN error (frame quality)
Elf tech #1: per-window energy/CI fronts

- What’s the best count action for a window?
  - A count action: determining (1) an NN and (2) # of frames to process

When energy is low: cheaper NNs win
When energy is low: more accurate NNs win

Energy/CI front: the combination of all “optimal” count actions with varied energy

- How to construct? Error integration
- Depends on the video characteristics
Elf tech #1: per-window energy/CI fronts

- What’s the best count action for a window?
  - A count action: determining (1) an NN and (2) # of frames to process

When energy is low: cheaper NNs win
When energy is low: more accurate NNs win

Energy/CI front: the combination of all "optimal" count actions with varied energy

- How to construct? Error integration
- Depends on the video characteristics
Elf tech #2: across-window joint planning

- An Oracle Planner: best performance but unrealistic
  - knows all energy/CI fronts

A greedy approach: giving energy to the window with the most benefit (i.e., CI width reduction).
Elf tech #2: across-window joint planning

- An Oracle Planner: best performance but unrealistic
  - knows all energy/CI fronts

A greedy approach: giving energy to the window with the most benefit (i.e., CI width reduction).
Elf tech #2: across-window joint planning

- An Oracle Planner: best performance but unrealistic
  - knows all energy/CI fronts

A greedy approach: giving energy to the window with the most benefit (i.e., CI width reduction).
Elf tech #2: across-window joint planning

- An Oracle Planner: best performance but unrealistic
  - knows all energy/CI fronts
Elf tech #2: across-window joint planning

- An Oracle Planner: best performance but unrealistic
  - knows all energy/CI fronts
- Challenges:
  - Needs global knowledge (budget planning)
  - On-the-go (cannot delay, needs to provide fresh data for users)
Elf tech #2: across-window joint planning

- An Oracle Planner: best performance but unrealistic
  - knows all energy/CI fronts
  - Planned offline

- A learning-based planner: imitating the oracle planner
  - basis: reinforcement learning
  - rationale: daily and temporal patterns
Elf tech #2: across-window joint planning

- An Oracle Planner: best performance but unrealistic
  - knows all energy/CI fronts
  - Planned offline

- A learning-based planner: imitating the oracle planner
  - basis: reinforcement learning
  - rationale: daily and temporal patterns
  - offline training -> online prediction
    - Two agents: NN selection and # of frames
    - Observations: knowledge of past windows
    - Penalty: deviation from oracle’s decision
Elf tech #2: across-window joint planning

- An Oracle Planner: best performance but unrealistic
  - knows all energy/CI fronts
  - Planned offline

- A learning-based planner: imitating the oracle planner
  - basis: reinforcement learning
  - rationale: daily and temporal patterns
  - offline training -> online prediction
  - Enforce energy budget:
    - Different models for different budget level
    - Backstop design in case of no-energy
      - make reservation for future windows
      - Tuning reward functions to be conservative on energy
Elf Implementation

- Capture & processing decoupled for higher energy efficiency
  - Processing batched at the end of each window
Elf Evaluation

- Over 1,000-hr videos
  - Public, 2-week long each stream

- Baselines
  - 1. GoldenNN: most accurate NN
  - 2. UniNN: one fixed best NN
  - 3. Oracle: offline planned

- Small solar panel
  - 10Wh~30Wh per day
Elf Evaluation

- Average: 11% error, valid and 17%-width CI
- 95% confidence level

Cis cover ground truth with 95% probability (specified)
Elf Evaluation

- Average: 11% error, valid and 17%-width CI
- Significant improvements over baselines in CI widths
  - The number in the table is how much Elf closer to oracle (best case)

<table>
<thead>
<tr>
<th>Budget (per day)</th>
<th>10Wh</th>
<th>20Wh</th>
<th>30 Wh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Golden NN</td>
<td>66.6%</td>
<td>59.8%</td>
<td>56.2%</td>
</tr>
<tr>
<td>UniNN</td>
<td>41.1%</td>
<td>16.6%</td>
<td>9.7%</td>
</tr>
</tbody>
</table>
Elf Evaluation

- Average: 11% error, valid and 17%-width CI
- Significant improvements over baselines in CI widths
  - The number in the table is how much Elf closer to oracle (best case)
  - Very close to oracle
Elf Evaluation

- Average: 11% error, valid and 17%-width CI
- Significant improvements over baselines in CI widths
- Very close to oracle
- What if we have AI accelerators?
  - CIs are reduced noticeably (by 22.1%–33.1%)
  - Still cannot process every frame (short of energy)
Summary

- Autonomous camera: expanding the geo-frontier of video analytics
  - Energy-independent and compute-independent

- Elf: the first runtime for autonomous camera
  - Target query: object counting
  - Key idea: count planning per- and across-windows

- Prototyped on heterogeneous hardware

- Evaluated on over 1,000-hr videos
  - 11% error, 17% CI width
Very high-level thoughts of the paper

- Pros:
  - Clear, precise problem definition, use case
  - Comprehensive discussion and consideration (design & expr)

- Cons:
  - Not exactly a bounded error. Error aggregation function looks empirical
  - Not sure how easy that can be applied to other query / problem
  - Experiments are still based on urban data

- Maybe a further topic
  - Duplicated / distinct object.
Moving computation to IoT devices

Reasons to keep it local / on-device:

- Device is becoming more powerful and chips are cheaper
- Privacy issue
- Data is becoming too large to transmit and compute with in-different cost
Challenges of IoT computation-related system

- Limited Bandwidth / Unstable wireless
  - 5g-IoT \(\text{survey}\)

- Limited energy

- Limited computational power

- Others:
  - Device heterogeneity
    E.g. **Heterogeneous Multi-Mobile Computing** (Mobisys 2019)
  - Mobile
  - Context-aware
Some big areas in IoT

- Security
  - Blockchain related (survey)

- Sensing related:
  - E.g. Localization

- Healthcare

- AI on the edge / device

- Application

- Network management, mmWave etc
Feeling some diff between system and IoT research

System: (a little more) driven by expectation (metrics)

IoT: (a little more) limited by constraints