Multi-level Feature-driven Storage Management of Surveillance Videos

Min-Han Tsai
Networking **M**ultimedia **S**ystems **L**ab
ISA, National Tsing Hua University

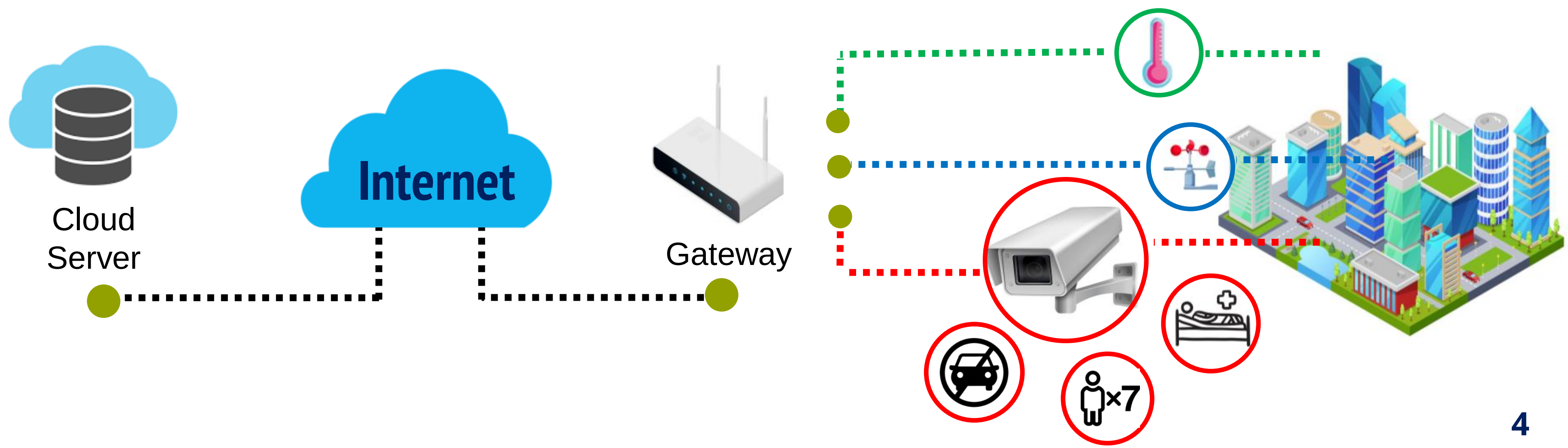
Outline

- Motivation
- Goals & Challenge
- Research Problems
- System Architecture
- Sampling Length Estimator: SLE
- Downsampling Decision Maker: DDM
- Implementation
- Evaluations
- Conclusion & Future Work

Motivation 1

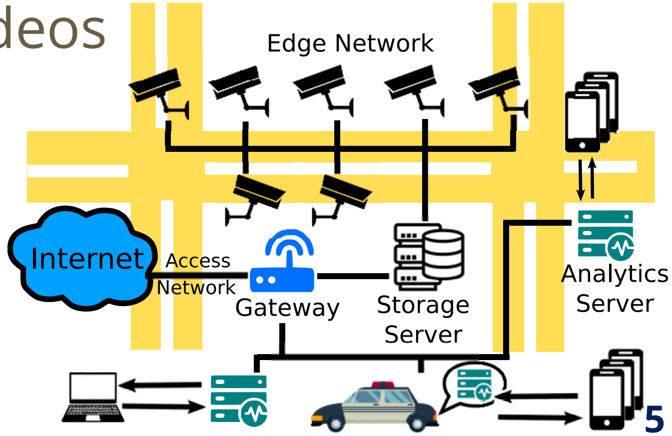
Motivation

- A smart city consists of many IoT devices with cameras, air quality sensors, thermometers, etc.
- These IoT devices benefits citizens and environments
- Among them, **surveillance cameras** become popular for
  - Tracking people, monitoring patients, detecting illegal parking
- Thousands of cameras are installed to provide seamless analytics
- Upload all videos to cloud directly leads to **network congestion**



Motivation

- One possible solution is to
    - Store video clips locally on an edge storage server
    - ➡ *Reduce the traffic load on access network*
    - End users can analyze videos for useful information no matter where they are
- But storage server has limited storage and computing power!
    - Fill up disks quickly, e.g., 1 Mbps video clips from 10 cameras in 1 week result in 1.4 TB data size
- To make room for incoming videos
    - Get rid of some videos or reduce sizes of videos
    - But, we want to retain informational videos
- How can we retain the most information amount under the limited storage space?



What is
Information Amount

What is Information Amount?

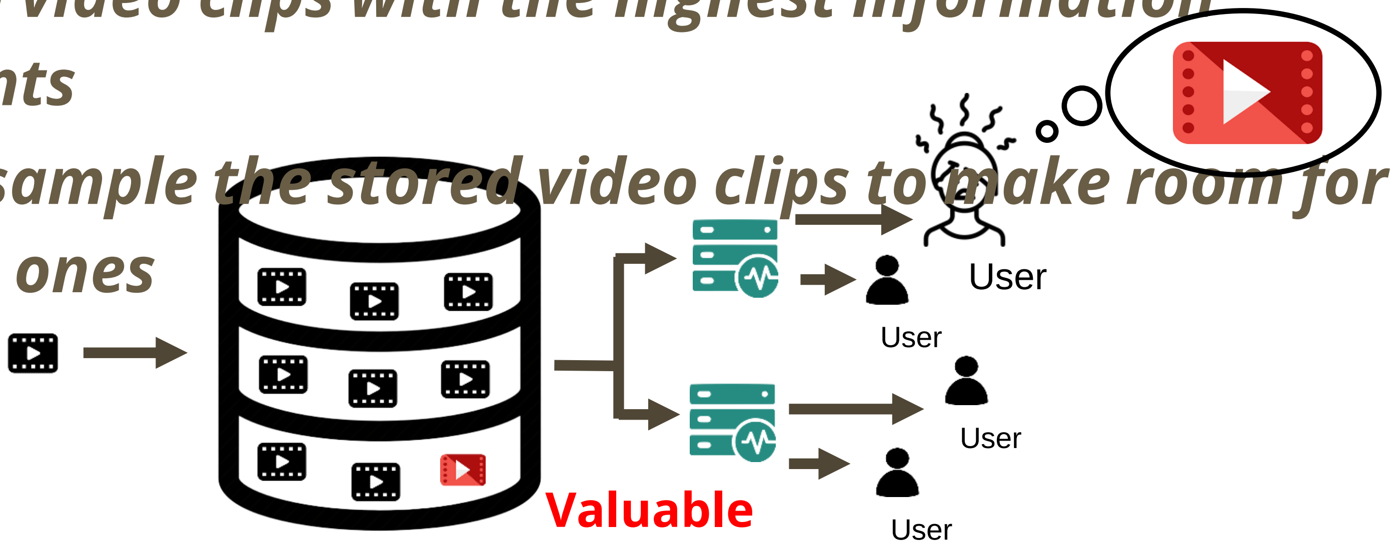
- Values of videos that depends on:
  - Analytic results from end users' needs
  - i.e., no. people, duration of illegal parking, or running red lights

Goals & Challenges 2

What Are Our Goals?

- Intuitive storage strategy:
  - Preserving videos with less or no information wastes storage space
  - FIFO loses too much information of videos

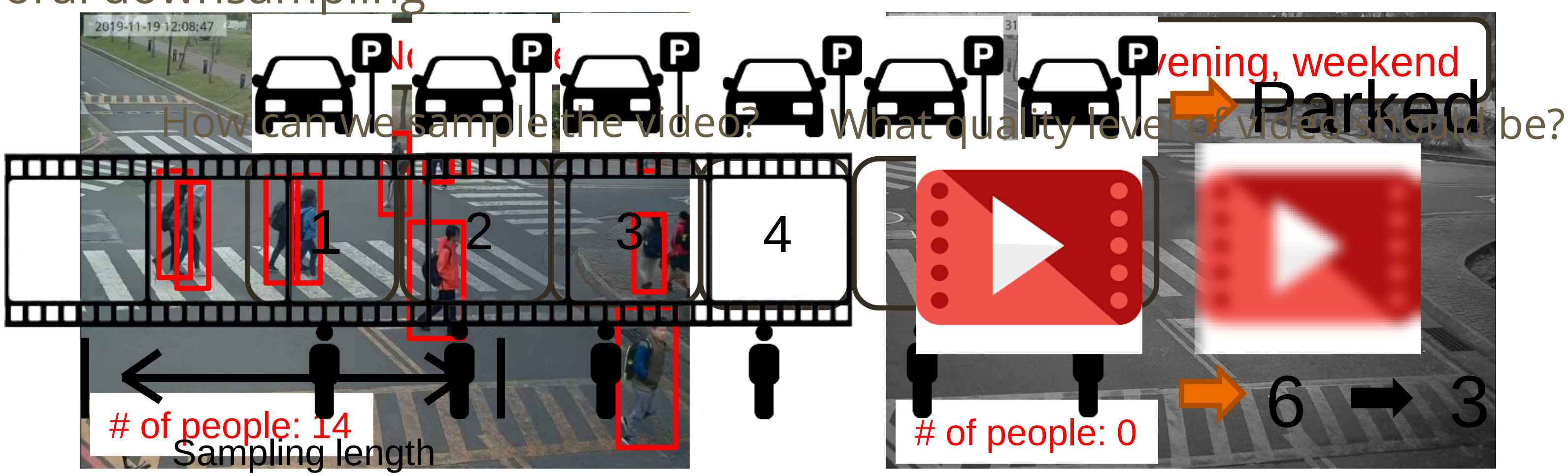
- **Our goals:**
  - ***Retain video clips with the highest information amounts***
  - ***Downsample the stored video clips to make room for future ones***



Challenges

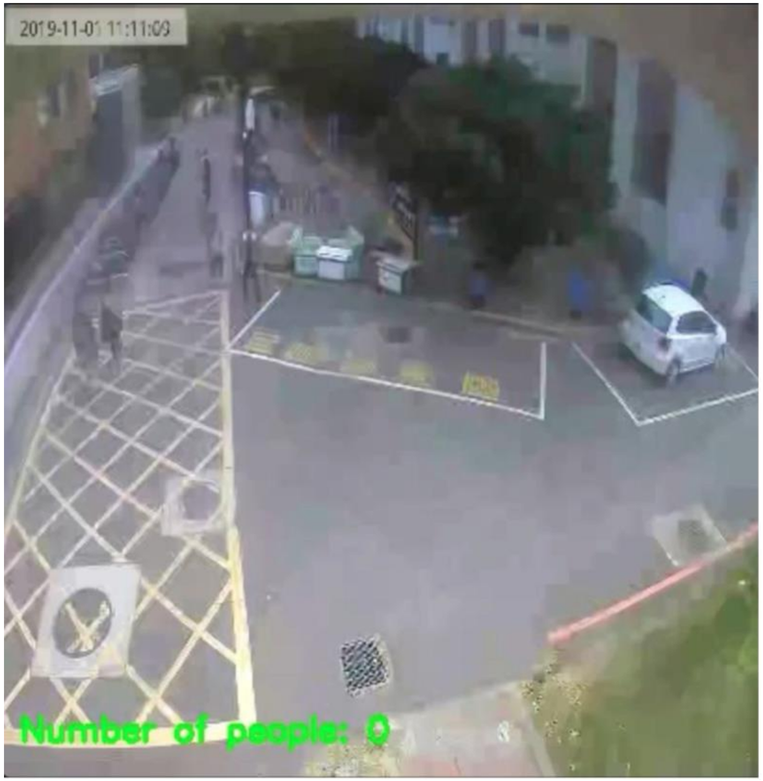
- Different video clips contain diverse information amounts
  - Depend on video analytics
- Different downsampling approaches lead to diverse information loss
  - Depend on video transcoders
- Quantifying the information amounts and downsampling video clips are both computationally intensive
  - Need to be carefully scheduled

Temporal downsampling



High Quality	Low Quality
(24 fps, 1000 kbps)	(1 fps, 10 kbps)

484.9 MB



3.8 MB

Lower video quality negatively affects the analytic results, but saves more storage space

3 Related Work

Video Summarization

" Video summarization produces a condensed and succinct representation of video content, which facilitates the browsing, retrieval, and storage of the original videos. " *[TMM'10]*

Keyframing

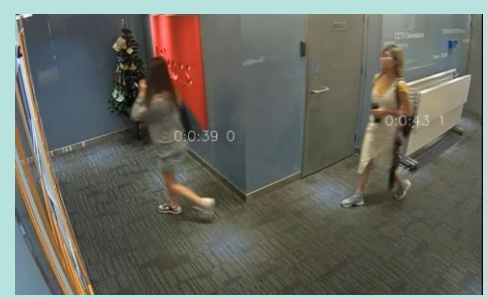
[JIOT'19, JVCIR'17]

- Composed of a set of **frames** extract from the original videos
- Not restricted by timing or synchronization issues
- More flexible for browsing

Video skimming

[TMM'10, AVSS'16]

- Composed of a set of **shots**
- Generated by considering the similarity or feature relationship among shots
- More intact for conveying information



https://www.youtube.com/watch?v=gk3qTMlcadk&t=166s

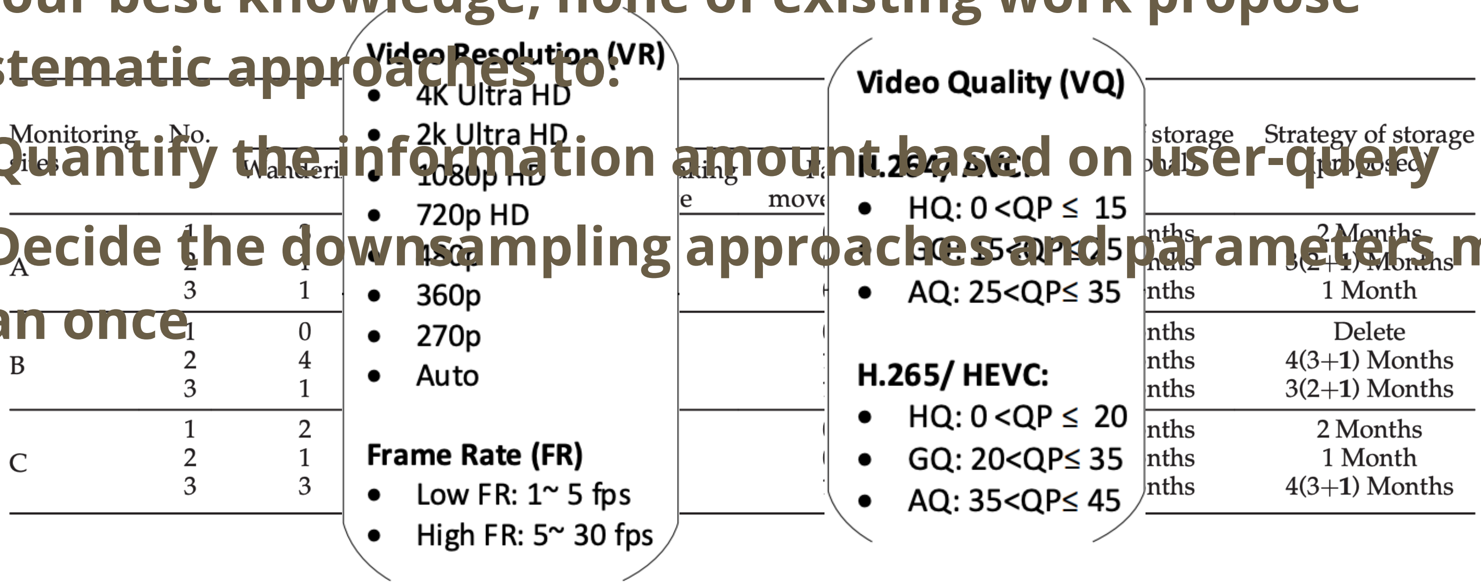
Video Server

- Usman et al. [INC'18]
  - Study correlation among the cameras at different locations
  - Encode video clips with different encoding parameters
  - Determine each clip to be deleted, partially deleted, or kept
- Shao et al. [TBD'18]
  - Build an intrusion driving model
  - Build a risk table
  - Downsample each video clip only once

To our best knowledge, none of existing work propose systematic approaches to:

1. Quantify the information amount based on user-query

2. Decide the downsampling approaches and parameters more than once



Contributions

1. Propose a storage sever to retain information amount under the constraints of space and computation power

2. Decide the sampling lengths for analytics and quality levels for preservation

3. Give optimal and approximate algorithms with analysis, and heuristic algorithms for better efficiency and practicality

4. Evaluate the performance of system in the real world testbed

System Overview 4

System Overview

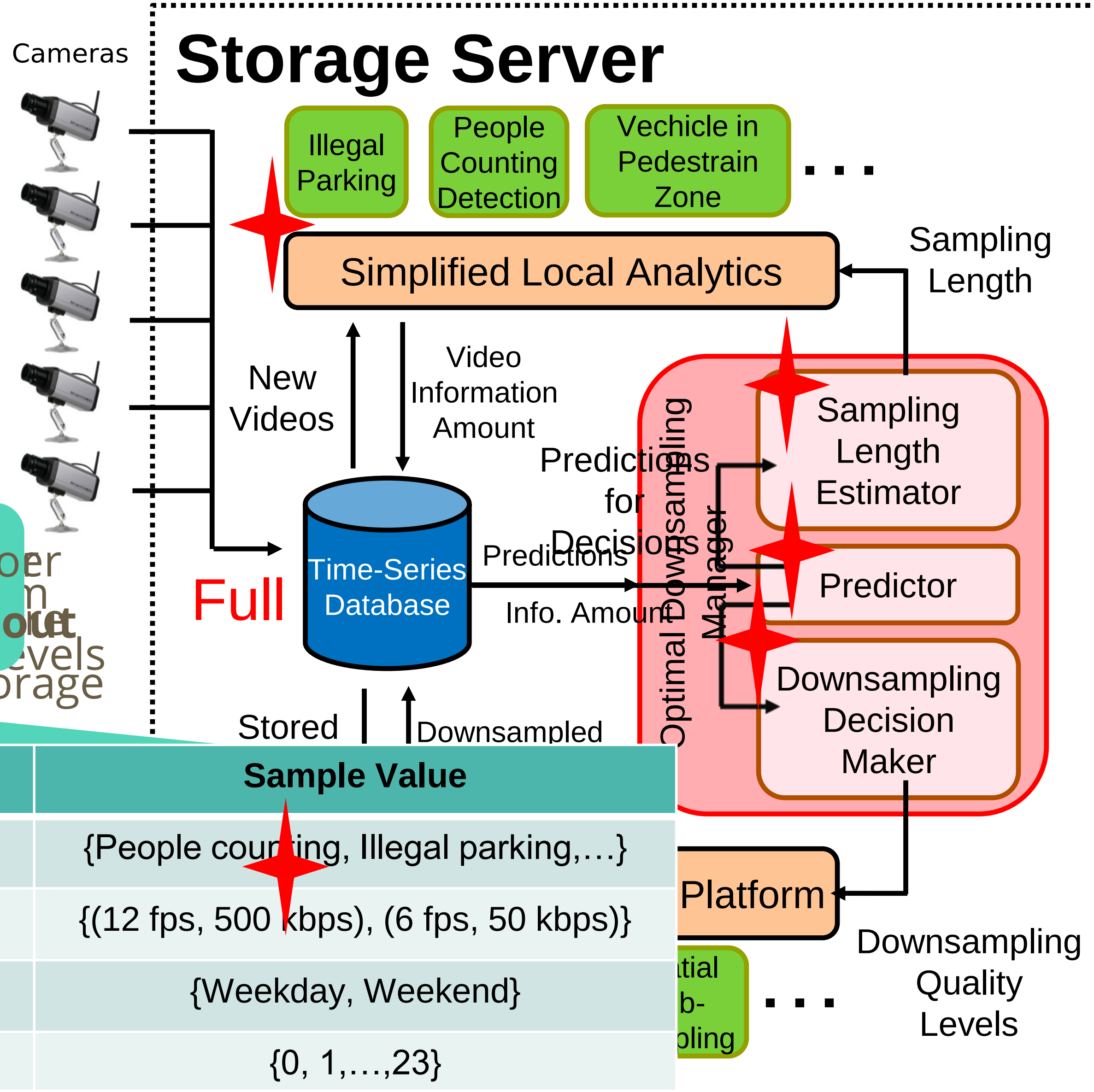
- Predictor
  - Lookup tables
  - Predictions for deciding sampling lengths and quality levels

Sample outputs

- Sampling Length Estimator
  - Decides sampling length for analyzing each video without overloading the server

Execution time: 17s
Resulting size: 65 MB
Information amount: 0.66

to downsample videos



Index	Sample Value
Analytics	{People counting, Illegal parking,…}
Downsample Decision	{(12 fps, 500 kbps), (6 fps, 50 kbps)}
Day-of-the-week	{Weekday, Weekend}
Time-of-the-day	{0, 1,…,23}

Research Problems 5

Information Amount
Importance of Videos

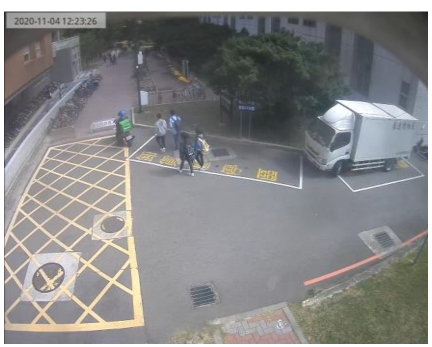
Visual Feature

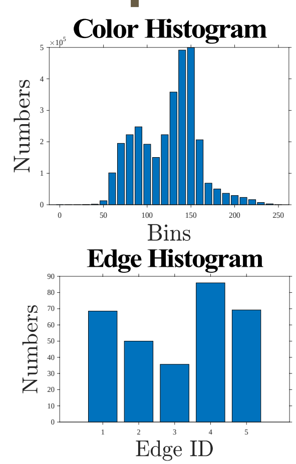
Semantic Feature

Characterize diverse amount of information

Information Amount Estimation

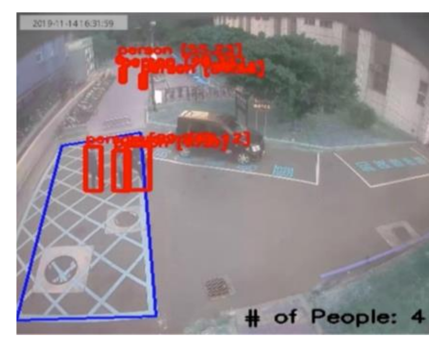
- Visual Feature:
  - Low-level and general across queries with heterogeneous analytics
  - E.g., color histogram, dominated edges, convolution……
  - Simpler and faster
  - **No need to be sampled**
- Semantic Feature:
  - High-level and directly reflect the user intended queries
  - E.g., duration of illegal parked, no. of people pass by
  - Resource starving and user-demanded
  - **Need to be sampled**





…

Illegal Parked: True



No. of People: 4

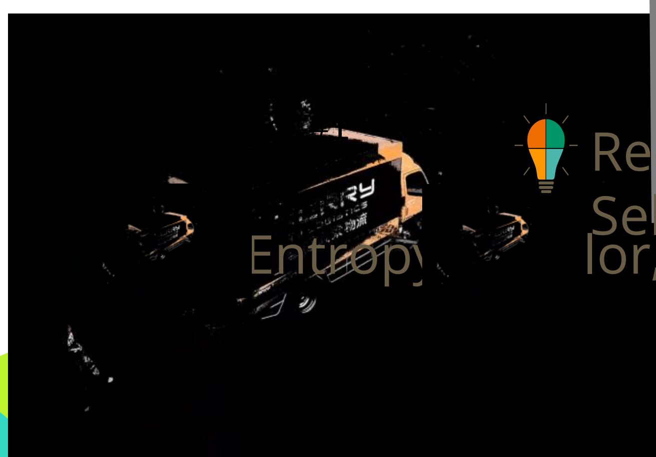
Visual Features Extraction

Remove the redundant content

Preserve important segments

$$I_v(S_c) = P_1([E(F_1), E(F_2), \ldots, E(F_m)])$$

Background Subtraction → **Shot Detection** → **Feature Extraction** → **Principal Component Analysis**



Reduce the dimension of features

Select the first component

Entropy lor, Edge and Ti ies

Semantic Feature Extraction

$$e_{S_c,a} = \begin{cases} 0 & |x_a - n_a| \leq \delta_a; \\ |x_a / \tilde{x}_a| & \text{otherwise,} \end{cases}$$

$$I_e(S_c) = \sum_{a \in A_c} \mathbf{w}_{c,a} \cdot e_{S_c,a} \Bigg/ \sum_{a \in A_c} \mathbf{w}_{c,a}$$

- x_a: output of analytics (boolean or integer)
- n_a: normal output
- $\tilde{x}_a$: maximal absolute value
- δ_a: semantic threshold
- $e_{S_c,a}$: information amount of analytic a in shot S_c
- $I_e(S_c)$: semantic info. amount in shot S_c

Total Information Amount

- Info. amount of new coming video clips
- Without sampling
- δ_v: visual threshold

$$\boxed{I(c, f_c)} = \sum_{S_c \in c} \hat{I}_v(S_c) + \sum_{\substack{\forall S_c^{'} \in c, \\ I_v(S_c^{'}) > \delta_v}} \hat{I}_e(S_c^{'}),$$

Consider all frames in the video clips

$$H(\mathbf{C}, \mathbf{F}, \mathbf{A}) = \sum_{c=1}^{\mathbf{C}} I(c, f_c)$$

Sampling Length Estimator: SLE

6

Sampling Length Estimator (SLE)

Optimal Downsampling Manager

Sampling Length Estimator

Predictor

Downsampling Decision Maker

- Approximate information amount

$$H'(\mathbf{L}) = \sum_{c=1}^{|\mathbf{C}|} I(c, L_c), \forall L_c \in \mathbf{L}_0$$

$$I(c, L_c) = \boxed{\sum_{S_c \in c} \hat{I}_v(S_c)} + \boxed{\sum_{\substack{\forall S_c^{'} \in c, \\ I_v(S_c^{'}) > \delta_v}} \hat{I}_e^{'}(S_c)}$$

- **Visual** info.: light weight, **no need** to be sampled/degraded
- **Semantic** info.: amount, **need** to be sampled/degraded

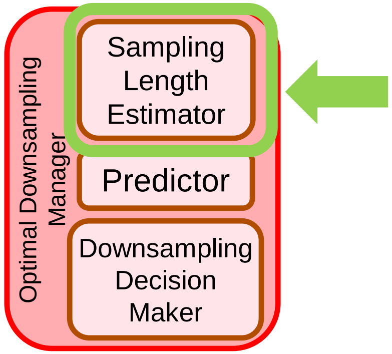
Degradation factor

$$I_e^{'}(S_c) = \frac{\sum_{a \in A_c} \mathbf{W}_{c,a} \cdot \hat{e}(c, f_c, a) \cdot \frac{|S_c|}{|\sum_{S_c \in c} S_c|} \cdot \boxed{d(c, a, L_{c,a})}}{\sum_{a \in A_c} \mathbf{W}_{c,a}}$$

$$= \frac{Sampled\ info.}{Complete\ info.}$$

- $L_c = (L_{c,a_1}, L_{c,a_2}, \ldots)$, pick up a frame from every L_c ones
- $\hat{e}(c, f_c, a)$: prediction of the information amount from unsampled video ($L_{c,a} = 1$)
- $W_{c,a}$: user-configured weight of analytic a of clip c

Sampling Length Estimator (SLE)



- Problem Formulation
  - Make approximate information $H'(L)$ as close to full-quality clips $H(C, F, A)$
  - Find the best L to analyze videos clips to maximize approx. info. amount

$$\min_{\mathbf{L}} \left(H(\mathbf{C}, \mathbf{F}, \mathbf{A}) - H'(\mathbf{L})\right) = \max_{\mathbf{L}} \left(H'(\mathbf{L})\right)$$

$$s.t. \quad \sum_{\forall c \in \mathbf{C}} \sum_{\forall a \in \mathbf{A}} \left(t(c, a) \cdot |L_{c,a}|\right) < \delta_i.$$

  - $t(c, a)$: execution time per frame when executing analytic a on clip c
  - δ_i: time constraint

Our SLE problem is NP-Hard reduced from Multiple Choice Knapsack Problem (MCKP)

➡ We propose optimal (OE), approximated (AE), and efficient (EE) algorithms

Optimal Estimation (OE)

- Dynamic programming based solution
  - Let $z(c, \delta)$ to be the maximal information
    - Considering the first $|c|$ clips under time constraint δ
    - The state of recursion is written as:

$$z(c, \delta) = \max\left(z(c-1, \delta - \sum_{\forall l_{c,a} \in l_j} t(c,a) \cdot l_{c,a}) + I(c, l_j)\right), \forall l_j \in \mathbf{L}_0,$$

- Lengths are found from pre-selected and discrete set $\boldsymbol{L}_0$
- The optimal solution is found at $z^* = z(|\boldsymbol{C}|, \delta_i)$
- Total time complexity: $O(|\boldsymbol{L}_0|\delta_i|\boldsymbol{C}|)$

  space complexity: $O(|\boldsymbol{C}|^2\ \delta_i)$

Approximated Estimation (AE)

- Binary-search based (*branching*) solution
  - Determine the optimal solution $z^* < x(1+\epsilon)$ or $z^* > x(1-\epsilon)$ exists
  - With ϵ=0.6, AE makes $z^*/\ z^0 \leq 5$ [IPL'98, vol. 67]
  - Total time complexity: $O(|\boldsymbol{C}||\boldsymbol{L}_0|log|\boldsymbol{C}|)$

Algorithm 1 Approximate Estimation (AE) Algorithm for the SLE Problem

Inputs: Clips $\mathbf{C}$, Deadline δ_i, Approximate Sampling Lengths $\mathbf{L}_0$, and Predictor $\hat{e}(\cdot)$
Output: Approximate Sampling Matrix $\mathbf{L}_x$.

1: Let $B_l = \max_{\forall c \in \mathbf{C},\ l_j \in \mathbf{L}_0} (I(c, l_j))$, $B_u = |\mathbf{C}| \cdot B_l$, and $\epsilon = 0.6$

Initialize Upper/Lower bounds

2: $x = B_u \ / \ 2$
3: $\mathbf{J} = \emptyset$
4: **for** $c \in \mathbf{C}$ **do**
5: $\{\|I\|\} = \frac{I(c,\ l_j)}{\sum_{\forall a \in A_c} (t(c,a) \cdot l_{c,a})}$, $\forall l_k \in \mathbf{L}_0 \ \cap \ \|I\| > \frac{0.8x}{\delta_i}$

Pick up the lengths that meet the constraint

6: $l_k = \arg\max_{l_j} (\|I\|)$
7: $\mathbf{J} = \mathbf{J} \cup \{(c, l_k)\}$
8: $\|\mathbf{J}\| = \sum_{\forall (c,l_k) \in \mathbf{J}} I(c, l_k)$
9: **if** $\|\mathbf{J}\| \leq 0.8x$ **then**

Check the ratio of bounds and adjust until guaranteed error

10: $B_u = x(1+\epsilon) = 0.8B_u$
11: **else**
12: $B_l = x(1-\epsilon) = 0.2B_u$
13: **if** $B_u \ / \ B_L \leq 5$ **then**
14: Construct $\mathbf{L}_x$ by $\mathbf{J}$
15: return $\mathbf{L}_x$
16: **else**
17: $x = B_u \ / \ 2$ and go to line 3

G. Gens, E. Levner, An approximate binary search algorithm for the multiple-choice knapsack problem, Information Processing Letters 67 (1998) 261–265.

Efficient Estimation (EE)

- Greedy based solution
  - Intuition: *execution time and accuracy of information amount are both reduced once the sampling length is increased*
  - Keep checking total execution time until reaching the time constraint δ_i

Algorithm 2 Greedy Estimation (EE) Algorithm for the SLE Problem

Inputs: Clips $\mathbf{C}$, Deadline δ_i, Sampling Lengths $\mathbf{L}_0$, and Predictor $\hat{e}(\cdot)$
Output: Efficient Sampling Matrix $\mathbf{L}_e$.

Let $\mathbf{L}_c = 1, \forall c \in \mathbf{C}$
while $\sum_{\forall c \in \mathbf{C}} \sum_{\forall a \in \mathbf{A}} (t(c,a) \cdot |L_{c,a}|) > \delta_i$ **do**
    Find $(c,a) = \arg\min_{(c,a)} \left(\mathbf{W}_{c,a} \cdot \hat{e}(c, f_c, a) \cdot \frac{|S_c|}{|\sum_{S_c \in c} S_c|} \cdot d(c, a, L_{c,a}) \cdot \frac{1}{t(c,a) \cdot |L_{c,a}|)} \right), \ \forall\ \mathbf{L}_c \neq 0$
    if **then**$L_{c,a} = \max(\mathbf{L}_0)$
        $\mathbf{L}_c = 0$
    else
        Let $L_{c,a}$ of $\mathbf{L}_c$ be the next larger length in $\mathbf{L}_0$
Construct L_e from selected sampling lengths
return $\mathbf{L}_e$

Find clip and analytics with maximal informtion amount per unit time

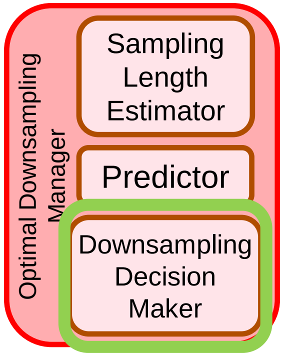
Update with a more light-weight sampling length

Time complexity: $O(\delta_i)$
Space complexity: $O(|\boldsymbol{C}||\boldsymbol{A}|)$

Downsampling Decision Maker: DDM

7

Downsampling Decision Maker (DDM)



- Downsampled information amount

$$H'(\mathbf{P}) = \sum_{c=1}^{|\mathbf{C}|} I(c, P_c)$$

Min-max normalized visual and semantic feature

$$I(c, P_c) = \sum_{S_c \in c} \hat{I}_v(S_c) + \sum_{\substack{\forall S_c' \in c,\\ I_v(S_c') > \delta_v}} \hat{I}_e''(S_c')$$

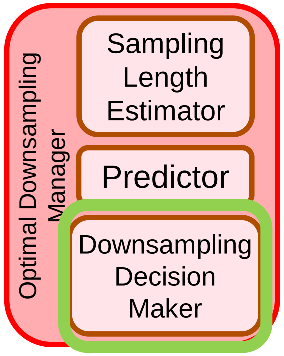
Degradation factor for downsampling

$$I_e''(S_c) = \sum_{a \in A_c} \mathbf{W}_{c,a} \cdot e_{c,a} \cdot \boxed{d'(c, a, P_c)} \Big/ \sum_{a \in A_c} \mathbf{W}_{c,a}$$

$$= \frac{Donwsampled\ info.}{Original\ quality\ info.}$$

- P_c: downsampling quality level of clip c
- $e_{c,a}$: captured info. amount from SLE
- $W_{c,a}$: user-configured weight of analytic a of clip c

Downsampling Decision Maker (DDM)



- Problem Formulation
  - Make downsampled information amount $\boldsymbol{H}'(\boldsymbol{P})$ as much as possible
  - Find the best quality P to store video clips

$$\max_{\mathbf{P}} (H^{'}(\mathbf{P})) = \max(\sum_{c=1}^{|\mathbf{C}|} I(c, P_c))$$

$$s.t. \quad \sum_{\forall c \in \mathbf{C}} t(c, P_c^{'}, P_c) < \delta_d, and \sum_{\forall c \in \mathbf{C}} \hat{o}_{c,P_c} < O_v.$$

- $t(c, P_c', P_c)$: downsampling time from quality P_c' to P_c
- δ_{d}: time constraint
- O_v: space constraint

Our DDM problem is NP-Hard reduced from Multi-dimensions Multiple Choice Knapsack Problem (MMCKP)

➡ We propose optimal (OD), approximated (AD), and efficient (ED) algorithms

Optimal Decision (OD)

- Dynamic programming based solution
  - Let $z(c, o, \delta)$ to be the maximal information, which considers
    - the first $|c|$ clips under space o and time constraint δ
    - Reach optimal solution at $z'(|\boldsymbol{C}|\ O_v\ \delta_d)$

$$z^{'}(c, o, \delta) = \max\left(z^{'}(c-1, o - \hat{o}_{c,p_j}, \delta - t(c, P_c^{'}, p_j)) + I(c, p_j)\right) \forall p_j \in \mathbf{P}_0,$$

    - Quality p_j: $(fps, bitrate, \ldots)$
    - Total time complexity: $O(|C|\ O_v\ \delta_d\ |P_0|)$

      space complexity: $O(|C|^2 O_v\ \delta_d)$

Approximate Decision (AD)

- Binary-search based (*branching*) solution
  - Keep adjusting upper/lower bound until the approx. solution falls in the range
  - With the approx. ratio at most $1 + 2d + (1/2)^{\hat{t}}$, where $d = \hat{t} = 2$ [RAIRO-Oper. Res.'16]
  - Total time complexity: $O(|\boldsymbol{C}|(t + \log(|\boldsymbol{C}| - 2d)))$, which is polynomial time

Algorithm 3 Approximate Decision (AD) Algorithm for the DDM Problem

Inputs: Information Amount I, Clips $\mathbf{C}$, Deadline δ_d, Approximate Downsampling Decision Matrix $\mathbf{P}_x$, Positive Integer $\hat{t}$.
Output: Approximate Downsampling Decision Matrix $\mathbf{P}_x$.

1: Let $B_l = \max_{\forall c \in \mathbf{C},\ p_k \in \mathbf{P}_0} (I(c, p_k))$, $B_{l_0} = B_l$, $B_u = |\mathbf{C}| \cdot B_l$, and $d = 2$
2: $x = \frac{d}{1+2d} B_u + B_l$
3: $\mathbf{J} = \emptyset$
4: **for** $c \in \mathbf{C}$ **do**
5: $\{\|I\|\} = \frac{I(c,\ P_c)}{t(c, P_c', P_c)/\delta_d} + \frac{I(c,\ P_c)}{\delta_{c,P_c}/O_v}$, $\forall\ p_i \in \mathbf{P}_0\ \cap\ \|I\| > \frac{x}{d}$
6: $p_k = \arg\max_{p_j} (\|I\|)$
7: $\mathbf{J} = \mathbf{J} \cup \{(c, p_k)\}$
8: $\|\mathbf{J}\| = \sum_{\forall (c,p_k) \in \mathbf{J}} I(c, p_k)$
9: **if** $\|\mathbf{J}\| \le \frac{1}{2d} x$ **then**
10: $B_u = (1 + \frac{1}{2d}) B_u$
11: **else**
12: $B_l = \frac{1}{2d} B_l$
13: **if** $B_u - (1+2d) B_l \le (\frac{1}{2})^{\hat{t}} B_{l_0}$ **then**
14: Construct $\mathbf{P}_x$ by $\mathbf{J}$
15: return $\mathbf{P}_x$
16: **else**
17: $x = \frac{1}{1+2d} B_u + d B_l$ and go to line 3

Based on dimension, we decide the upper/lower bounds

Pick the length that gives most information amount while meeting the constraint

Adjust the bounds until approx. solution falls into the range

C. He, J. Y. Leung, K. Lee, M. L. Pinedo, An improved binary search algorithm for the multiple-choice knapsack problem, RAIRO-Operations Research 50 (2016) 995–1001.

Efficient Decision (ED)

- Greedy based solution
  - Intuition:
    - *The video clip with the smallest per-unit-size information amount should be scarified first*
    - *Keep the degree of downsampling approach as small as possible*

Algorithm 4 Efficient Decision (ED) Algorithm for the DDM Problem

Inputs: Information Amount I, Weight $\mathbf{W}$, Deadline δ_d, Watermark O_v, Selected Sampling Length Matrix L, and Predictor $\hat{e}(\cdot)$.

Output: Efficient Downsampling Matrix P_e.

Let $\mathbf{P}_c = -1, \forall c \in C$; $S = \sum_{\forall c \in C} o_c$; $T = 0$;

while $S > O_v$ **or** $T > \delta_d$ **do**

$\quad c = \underset{\forall c \in C, \mathbf{P}_c \neq 0}{\arg\min} \; (I(c, L_c) \, / \, \hat{o}_{c,\mathbf{P}_c})$

$\quad \hat{p} = \underset{\forall \hat{o}_{c,\mathbf{P}_c} \geq \hat{o}_{c,\hat{p}}}{\arg\min} \; (\hat{o}_{c,\mathbf{P}_c} - \hat{o}_{c,\hat{p}})$

$\quad S = S - \hat{o}_{c,\mathbf{P}_c} + \hat{o}_{c,\hat{p}}$;

$\quad T = T - \hat{t}(c, \mathbf{P}_c) + \hat{t}(c, \hat{p})$;

$\quad \mathbf{P}_c = \hat{p}$

Construct P_e from the selected $\mathbf{P}_c$

return $\mathbf{P}_e$

Keep checking until space and time are acceptable

Get the video with most information per-unit-size

Estimatie used space based on the quality of video

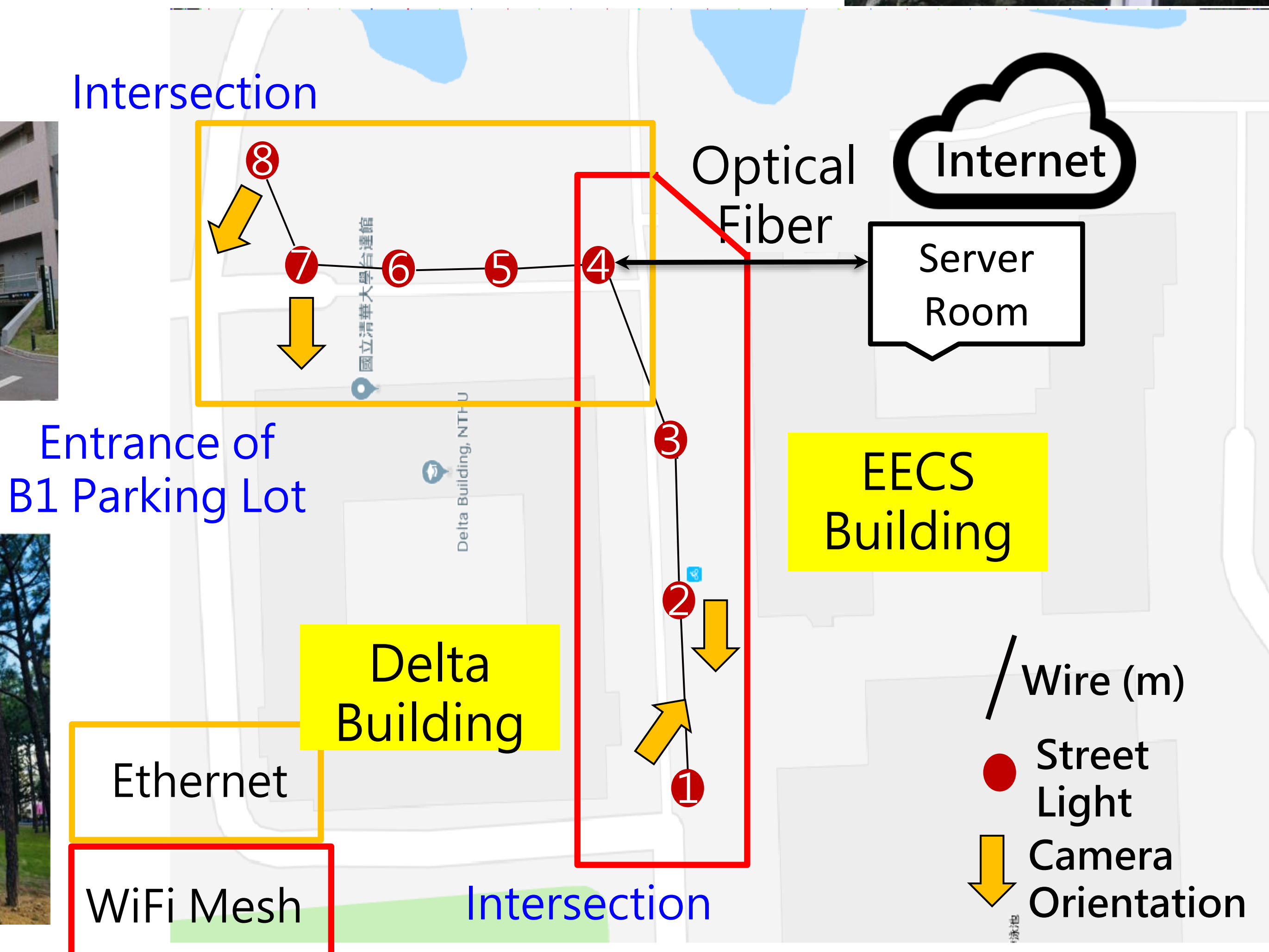
Time complexity: $O(S - O_v + \delta_d)$

Space complexity: $O(|\boldsymbol{C}|)$

Campus Testbed 8

Campus Testbed

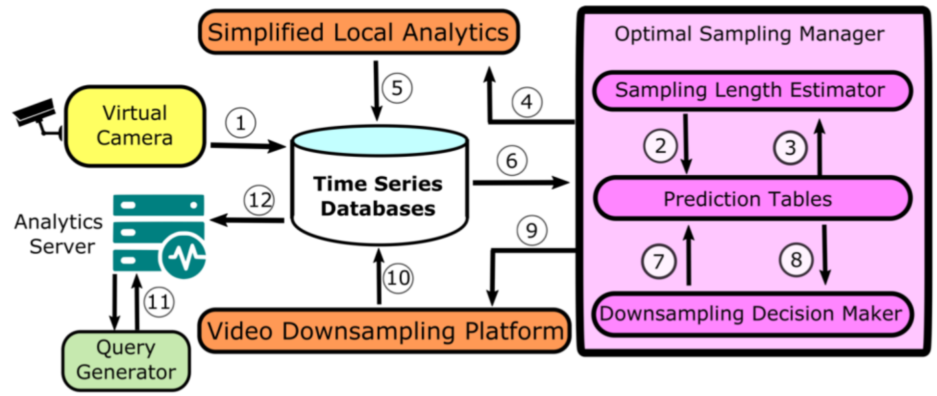
Thanks for the generous supports from LiteOn Inc.



Evaluations 9

Evaluation Setup (1/2)

- Current practices:
  - Equal-Fidelity (EF)
  - Equal-Frame-Rate (EFR)
  - First-In-First-Out (FIFO)



- The video clips are encoded
  - With HEVC, 1 Mbps, 24 fps, and 1 hour
  - From 12 continuous days in November, 2020
  - The first five days are warm-up
  - Sample results from a week, and we query (Poisson Process) on the last day

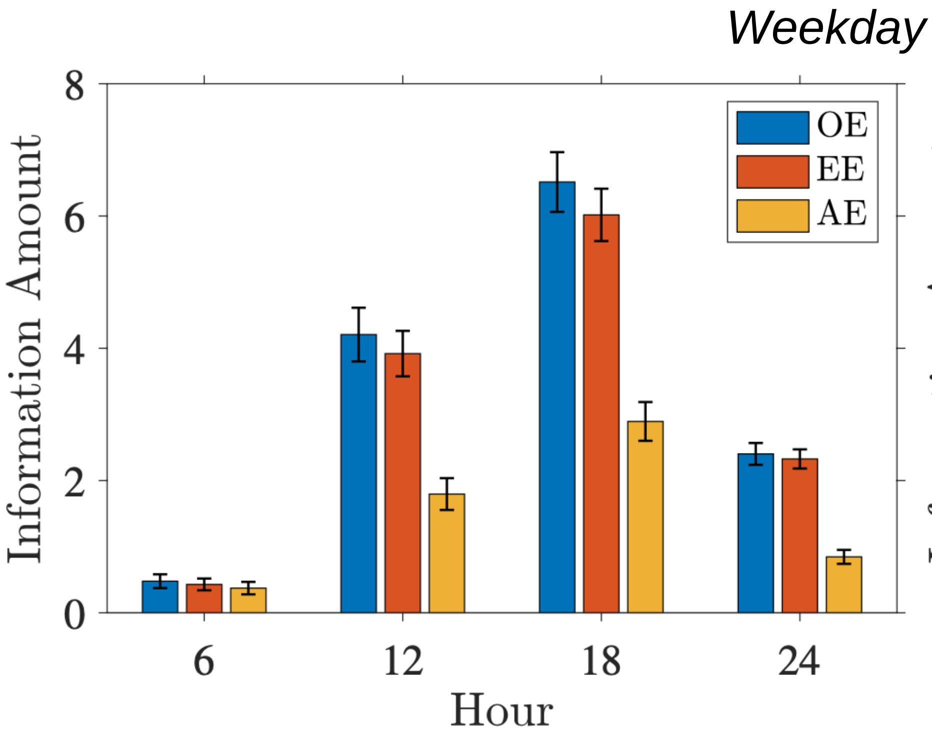
Evaluation Setup (2/2)

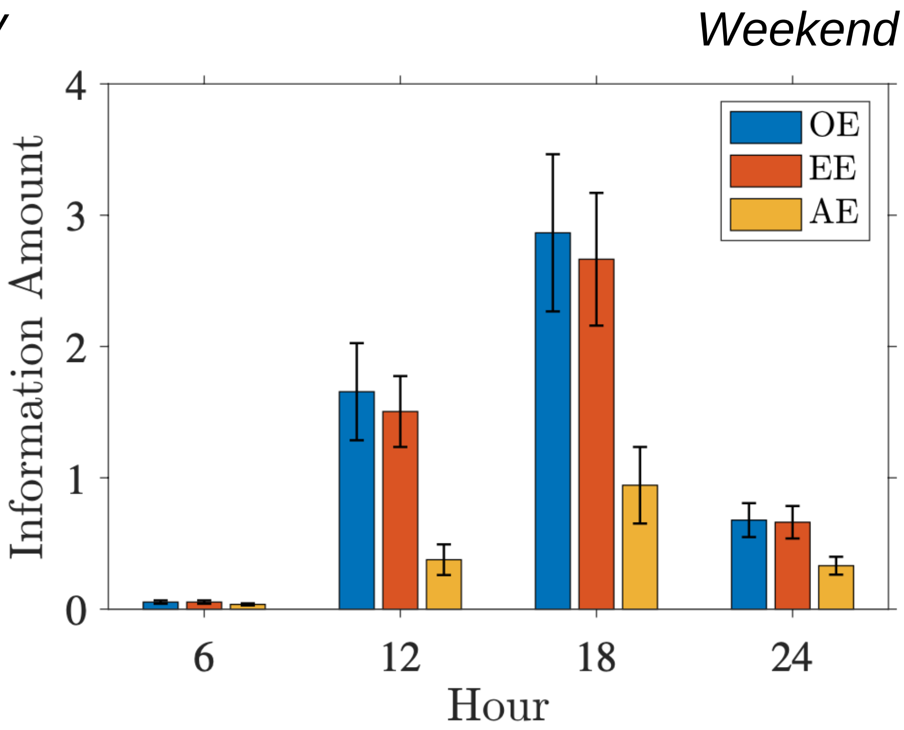
- Sampling length L_0: {1, 24, 48, 96, 144}
- Quality levels P_0: { (*24, 1000*), (24, 500), (12,500), (12,100), (6, 100), (6,10), (1, 10) }
- Analytics (known/unknown):
  - illegal parking#1, people counting, illegal parking#2, car counting
- Parameters:
  - Analysis deadline δ_i: 6 hours
  - Trigger SLE every 6 hours
  - Downsampling deadline δ_d: 6 hours
  - Storage space size O_v: {**20**, 40, 80 GB}
  - Watermark: Reduce 50% of size at least
  - Granularity levels: MB and GB
  - Error bar: 95 % confidence interval

Performance Metrics

- **Information amount**
  - SLE: estimated information amount over time
  - DDM: total information amount in storage server
- **Information amount error:**
  - User query (known/unknown)
- **Used storage space**
  - DDM: control of used space between watermarks
- **Number of stored video clips**
  - DDM : total number of clips stored in server (20/40/80 GB)
- **Running time of algorithms (OE/AE/EE, OD/AD/ED)**
  - SLE/DDM: analyzing/downsampling time of algorithms
  - SLE/DDM: running time of algorithms

Effectiveness of SLE Algorithms

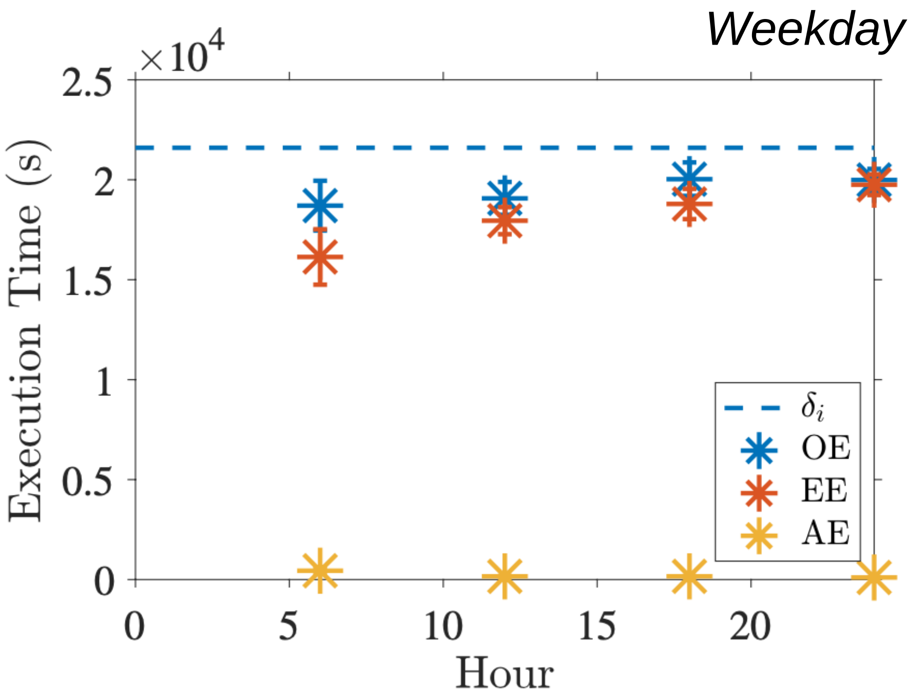




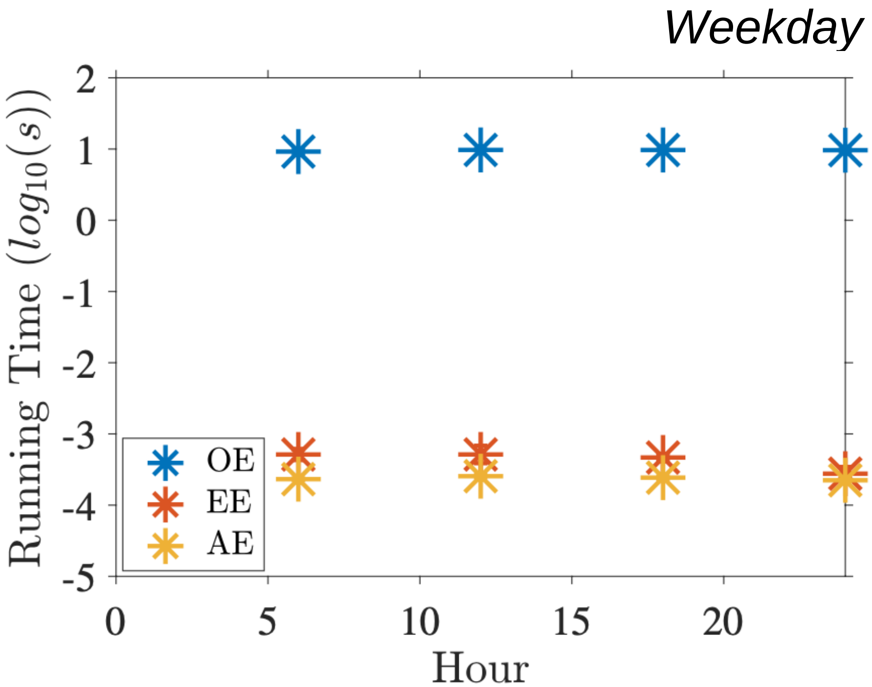
Our EE algorithm effectively estimates the sampling lengths for analyzing the videos, especially on peak time

Efficiency of SLE Algorithms

Analyzing time

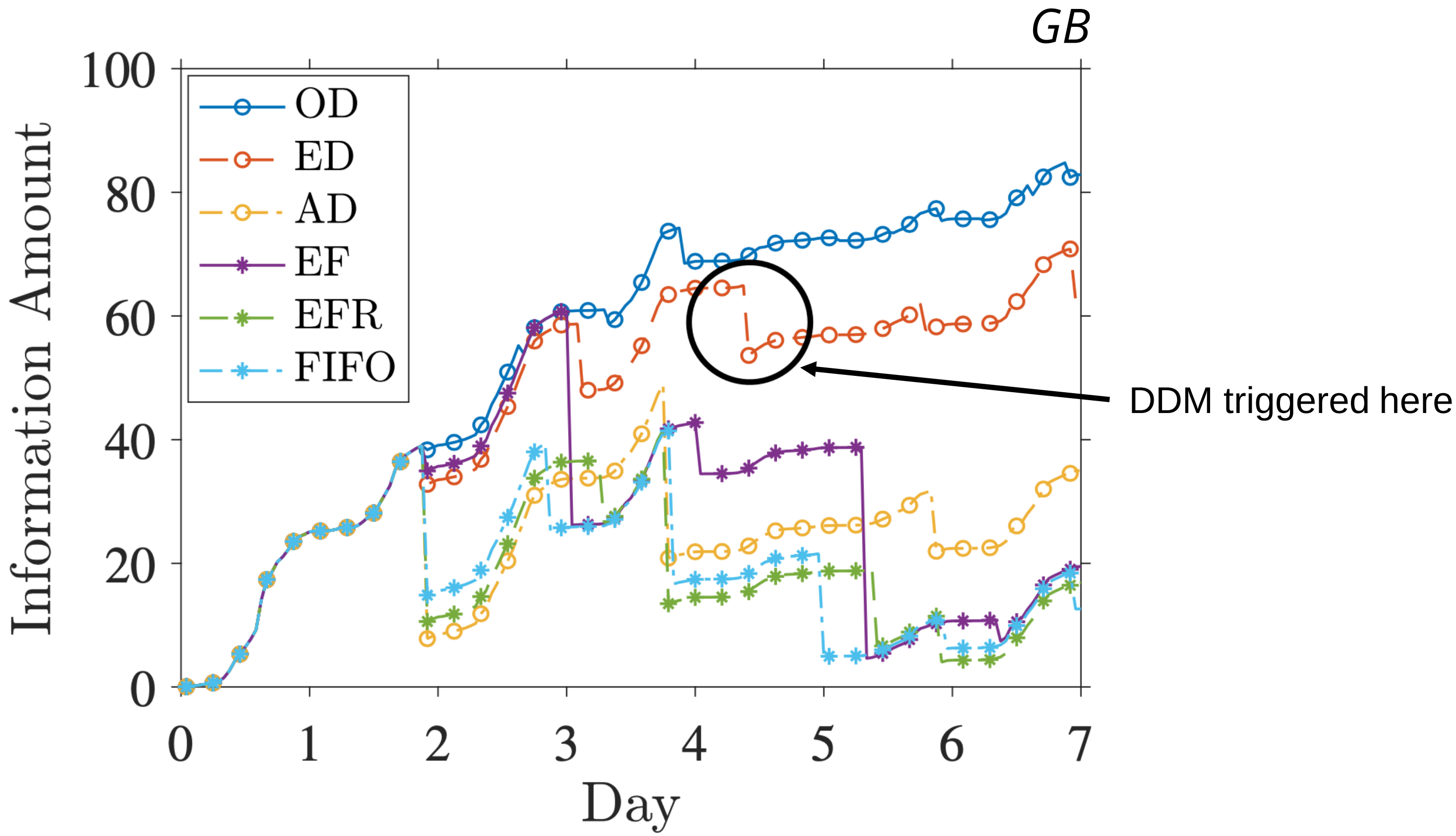


Algo. running time



We analyze all the videos in time
Our EE algo. runs in real time and faster than optimal over 10000 times

Total Information Amount on Storage Server



Our ED algo. outperforms AD by 44% and EF by 69%

Algo. Running Time & Granularity (1/2)

Running Time of Downsampling Decision Algorithm with different Granularity Levels

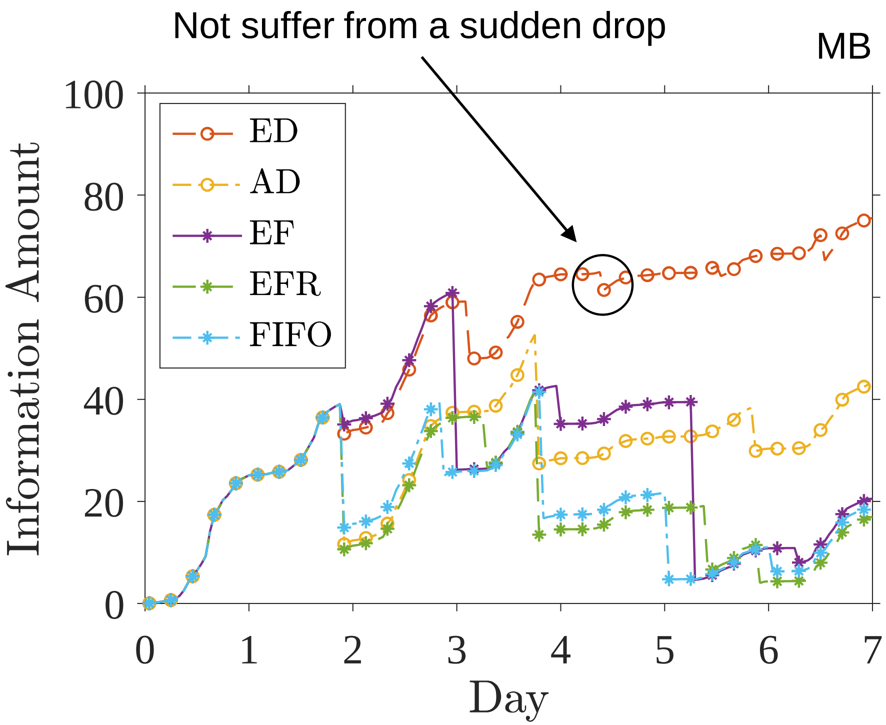
Algorithm	MB	GB
OD	N/A	$3.32 \times 10^{2} (\pm 1.95 \times 10^{1})$
ED	8.85×10^{-2} $(\pm 1.26 \times 10^{-3})$	1.25×10^{-2} $(\pm 2.72 \times 10^{-3})$
AD	2.06×10^{-3} $(\pm 2.28 \times 10^{-4})$	1.81×10^{-3} $(\pm 2.31 \times 10^{-4})$
EF	1.30×10^{-3} $(\pm 2.58 \times 10^{-3})$	1.00×10^{-3} $(\pm 2.64 \times 10^{-5})$
EFR	8.12×10^{-4} $(\pm 9.02 \times 10^{-5})$	9.13×10^{-4} $(\pm 1.05 \times 10^{-4})$
FIFO	$5.26 \times 10^{-4} (\pm 1.89 \times 10^{-5})$	$4.95 \times 10^{-4} (\pm 5.13 \times 10^{-6})$

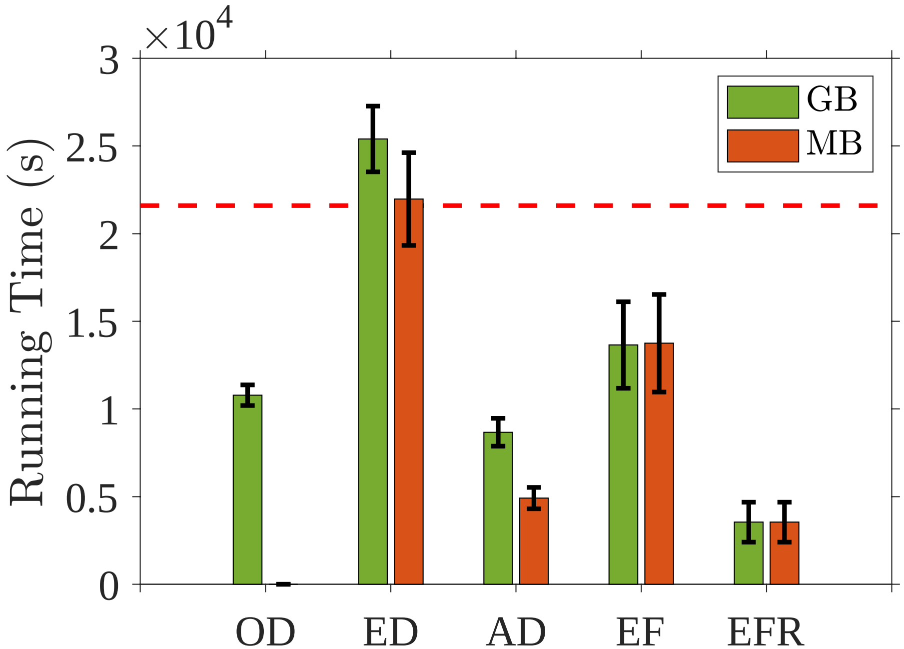
(Unit: s)

OD is not applicable in fine granularity

Our ED runs in real time in both fine/coarse granularity

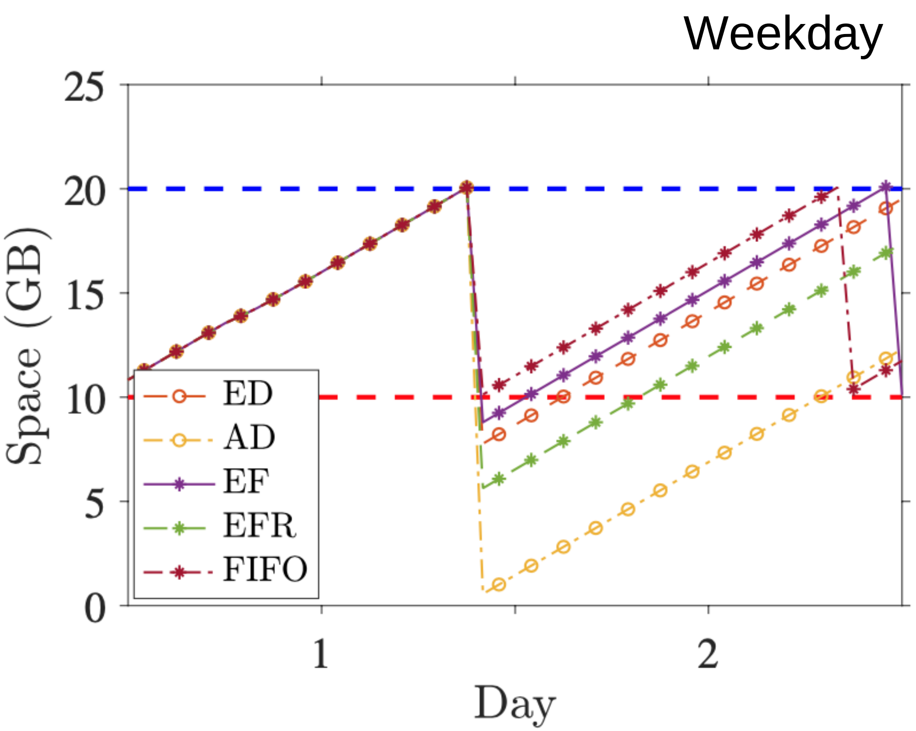
Task Running Time & Granularity (2/2)

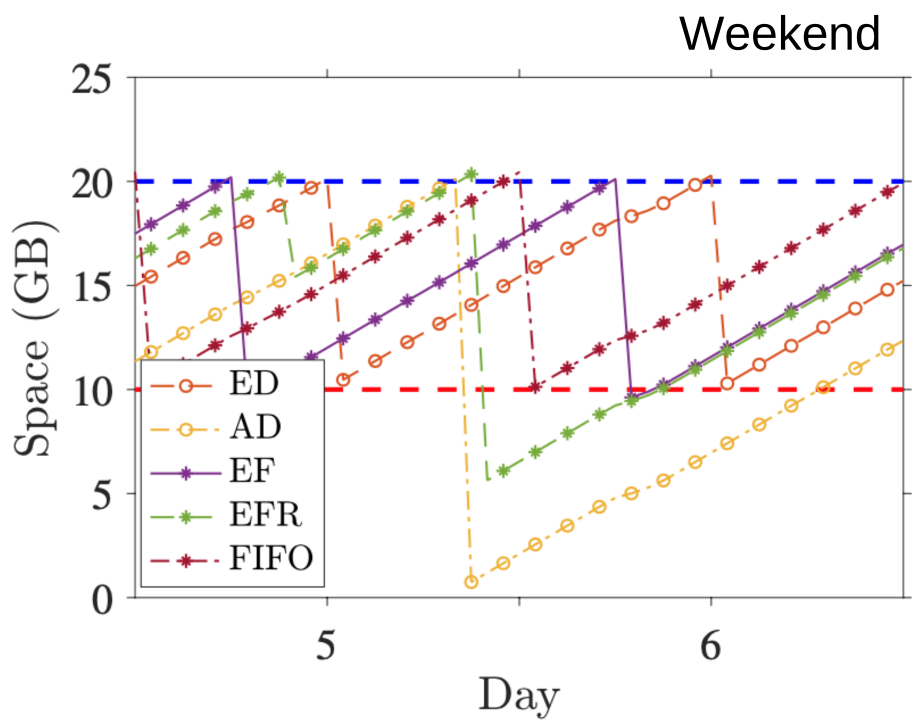




Our ED preserves more info. and meets dealine better under the fine granularity

Effectiveness of Storage Server

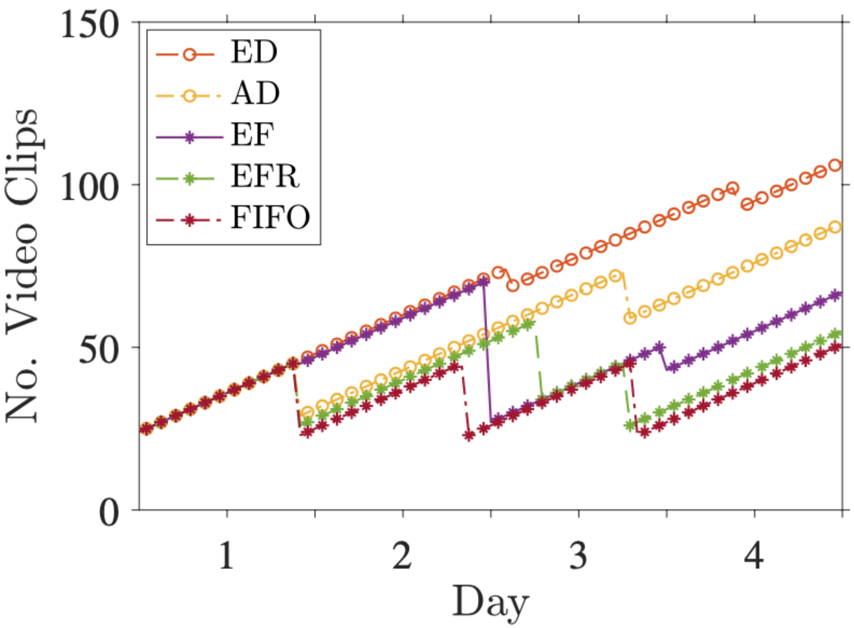




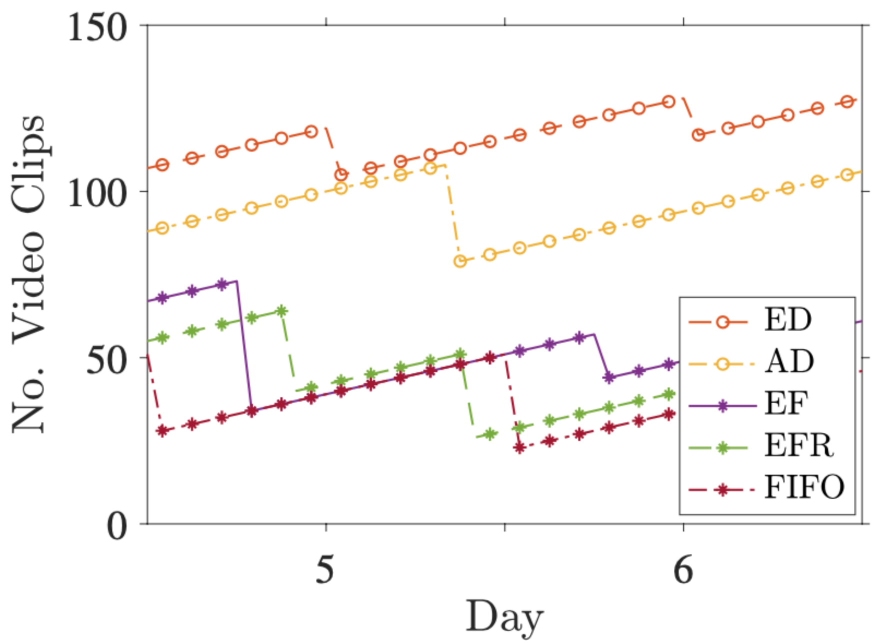
Our ED algo. manages the used space well on both weekday and weekend

Number of Video Clips in Storage Server

Weekday

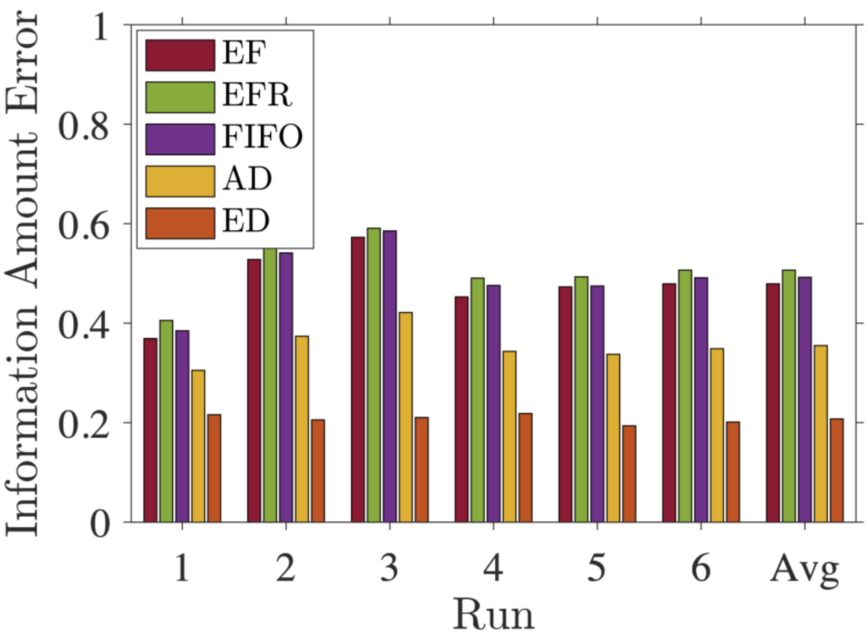


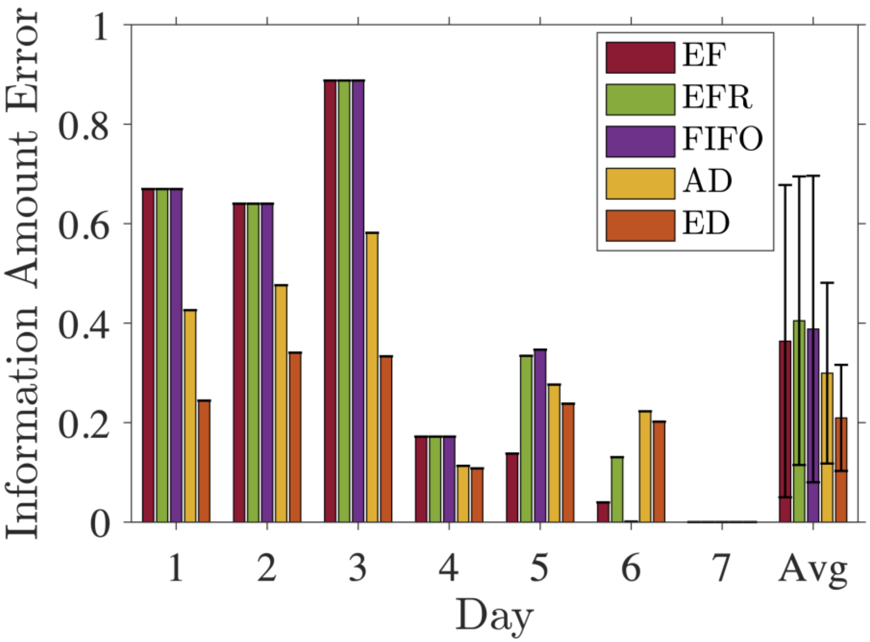
Weekend



Our ED removes 48% fewer clips than EF and saves 2.78 times more video clips than FIFO

Info. Error of Queries on the Last Day
(Known Analytics)





On average, the per-query error of our ED is 58% less than FIFO

Info. Error of Queries on the Last Day

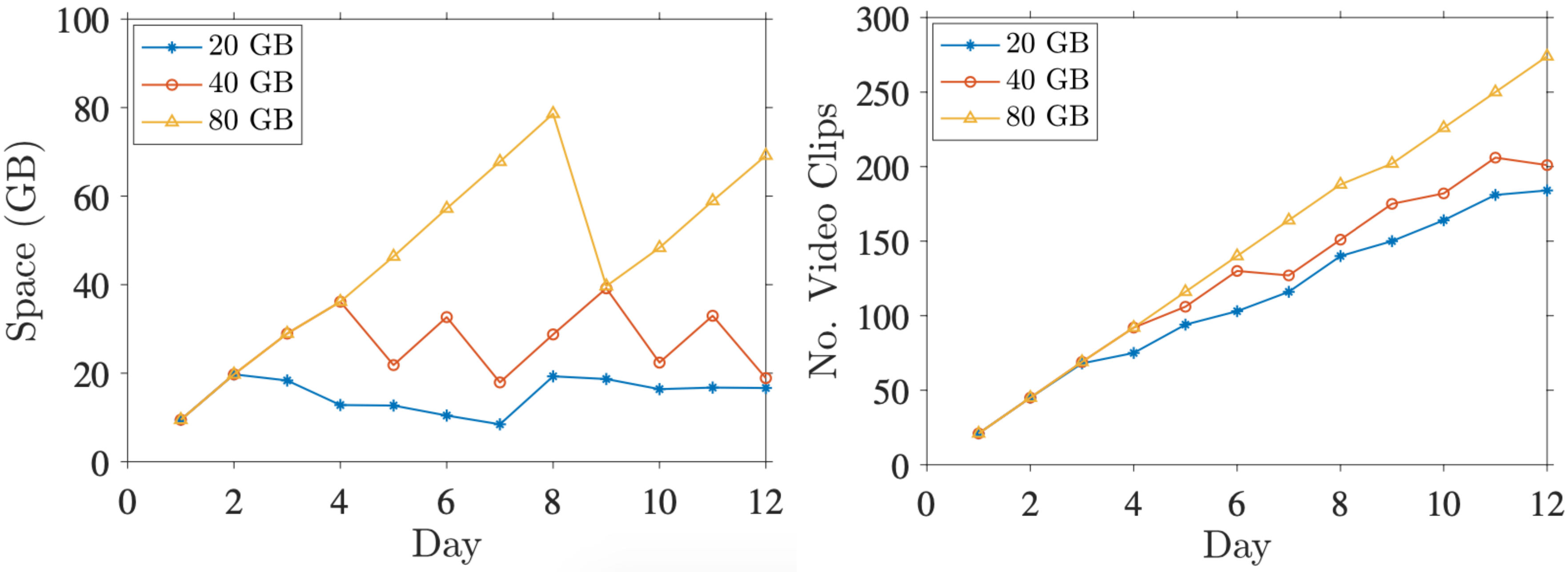
(Unknown Analytics)

Information Amount Error With and Without Visual Features

	Weekday	**Weekend**
With	9.77×10^{-2} $(\pm 1.14 \times 10^{-2})$	2.60×10^{-2} $(\pm 5.85 \times 10^{-3})$
Without	1.40×10^{-1} $(\pm 1.85 \times 10^{-2})$	4.78×10^{-2} $(\pm 1.03 \times 10^{-2})$

Introducing visual features leads to smaller information amount error: 30% on weekday and 46% on weekend

Performance With Larger Storage Space



Our ED successfully capitalizes additional storage space: used space is bounded between watermarks

Summary of Evaluations

- Our EE/ED algorithms look into the info. amount of unit time/space:
  - Achieve ~7% captured info. amount gap compared to the optimal
  - Boost the no. saved video clips by up to 2.78 times
  - Reduce per-query error by ~ 58% on average
  - Well-Manage the used space between watermarks
  - Scale well with larger storage space

Finish in time

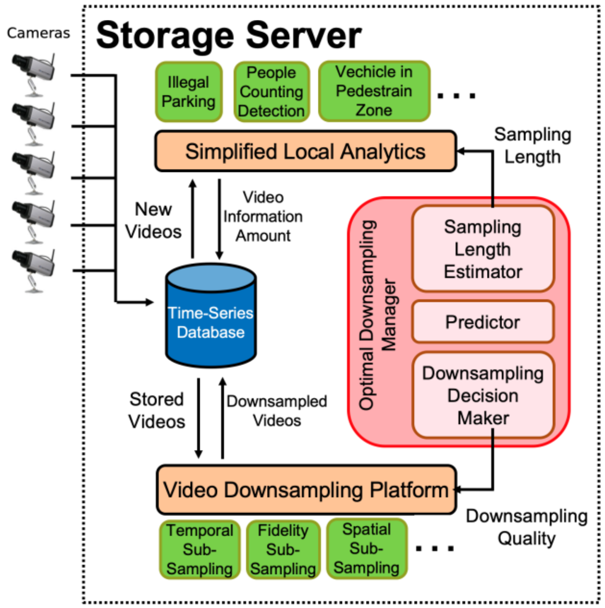
Preserve more information

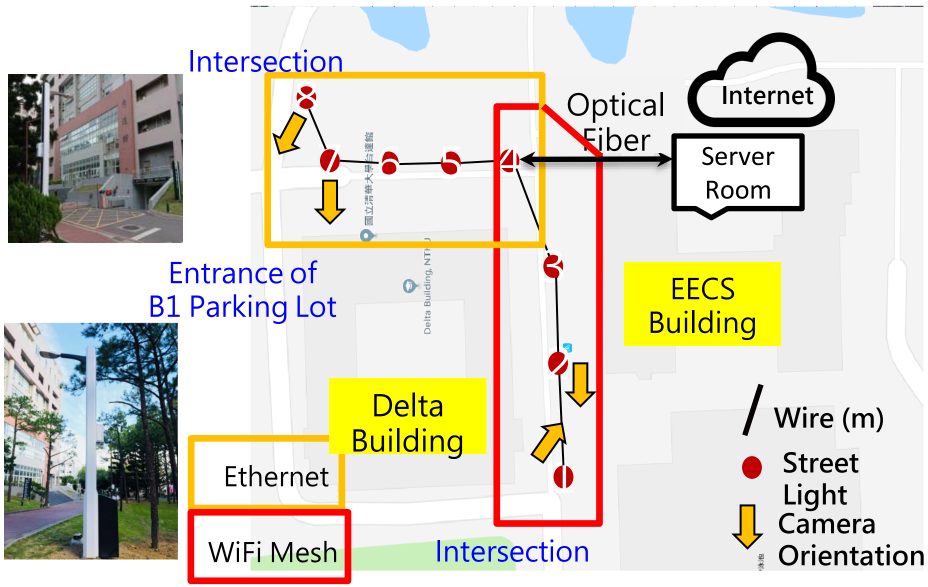
Well-manage storage space

Conclusion & Future Work 10

Conclusion

- We design, optimize, and implement a multi-level feature driven storage server for surveillance videos
  - Propose two algorithms (EE/ED) to determine the sampling lengths and stored quality levels of videos respectively
  - Evaluate our algorithms in a prototype implementation
  - Show our algorithms outperform the current practices





Future Work

- Build clusters of distributed storage server
- Incorporate the concept of Quality of Experience (QoE)
  - Reflect the real user satisfaction levels
- Apply more comprehensive predictions
  - E.g., Temporal regression, Reinforcement-Learning
- Consider a wider array of analytics
  - Information overlapped can be investigated in the storage server design

Publications

- M. H. Tsai, N. Venkatasubramanian, and C. H. Hsu, Multi-level Feature Driven Storage Management of Surveillance Videos, Journal of Pervasive and Mobile Computing, under review
- M. H. Tsai, N. Venkatasubramanian, C. H. Hsu, Analytics-aware storage of surveillance videos: Implementation and optimization, in: Proc. of IEEE International Conference on Smart Computing (SMARTCOMP), 2020, pp. 25–32.

THANKS!

Any questions?

mtsai.nthu@gmail.com