PacketGame:
Multi-Stream Packet Gating for Concurrent Video Inference at Scale

Mu Yuan, Lan Zhang, Xuanke You, Xiang-Yang Li

University of Science and Technology of China
Outline

• Background
  • PacketGame Design
• Evaluation
Background
Development Experience

• Video analytics system at University of Science and Technology of China
  • mobility analysis and anomaly detection
Background
Development Experience

• Video analytics system at University of Science and Technology of China
  • mobility analysis and anomaly detection
  • 1108 real-time 1080p streams from IP cameras
Background
Development Experience

• System Setup

  • 12 CPUs + TITAN X GPU edge server

  • YOLOX for object detection (on GPU)


![YOLOX Inference Diagram]

![Throughput (FPS) Graph]

Throughput (FPS)

Video Frames

YOLOX

Inference

27.7

10000

1000

100

10

1

YOLOX
Background
Development Experience

- Applying optimization techniques
- NVIDIA TensorRT (model inference acceleration)
Background
Development Experience

- Applying optimization techniques
- NVIDIA TensorRT (model inference acceleration)
- InFi (frame filtering, our MobiCom’22 paper)


Throughput (FPS)

<table>
<thead>
<tr>
<th>YOLOX</th>
<th>YOLOX +TensorRT</th>
<th>InFi</th>
</tr>
</thead>
<tbody>
<tr>
<td>27.7</td>
<td>753.9</td>
<td>3,589.4</td>
</tr>
</tbody>
</table>

YOLOX
Inference

Video Frames

InFi

Necessary Frames
passed by InFi
Background
Concurrency Bottleneck

- Concurrency benchmarks
Concurrent benchmarks

End-to-end concurrency is bottlenecked by the decoder (on 12 CPUs)
Background
Concurrency Bottleneck

- Concurrency benchmarks
- End-to-end concurrency is bottlenecked by the decoder (on 12 CPUs)
- Reason: all-frame decoding vs. partial inference
Background

New Idea

• Packet gating
  • selectively passing video packets to the decoder
  • reducing both decoder and inference overheads
Background

New Idea

- Packet gating
- selectively passing video packets to the decoder
- reducing both decoder and inference overheads

Comparison with Existing Ideas

<table>
<thead>
<tr>
<th>Methods</th>
<th>Reduce Decode</th>
<th>Commodity Cameras</th>
<th>Offline Videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Compression</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>On-Camera FF</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>On-Server FF</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Model Acceleration</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Packet Gating</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Outline

- Background
- PacketGame Design
- Evaluation
To selectively pass packets to the decoder, we need quantitive “scores” for video packets from concurrent streams.
PacketGame Design
Temporal Estimator

- Available hint#1: historical feedback
- Redundancy: the new inference result == the latest result
PacketGame Design
Temporal Estimator

- Available hint#1: historical feedback
- MAB-based approach
PacketGame Design
Temporal Estimator

- Available hint#1: historical feedback
- MAB-based approach
PacketGame Design
Temporal Estimator

• Available hint#1: historical feedback
• MAB-based approach
Packet Game Design
Temporal Estimator

- Available hint #1: historical feedback
- MAB-based approach
PacketGame Design
Temporal Estimator

- Available hint #1: historical feedback
- MAB-based approach

Stream-1
Stream-2
...  
Stream-N

Selected

Reward-1/N
1 for necessary
0 for redundant

Decoder

Model Inference

Video Frame-1/N

Selected

Inference

1 for necessary
0 for redundant
PacketGame Design

Temporal Estimator

- Available hint#1: historical feedback
- MAB-based approach

Stream-1

Stream-2

Stream-N
PacketGame Design
Contextual Predictor

• Available hint#2: packet-level metadata
• Metadata: packet size & picture type
PacketGame Design
Contextual Predictor

- Available hint\#2: packet-level metadata
- Metadata: packet size & picture type
- Neural network-based predictor

Two picture types: independent (I) & predicted (B/P)
Two sequences of packet sizes in a time window

Video Streams

Packet Parser → Packet Metadata → Contextual Predictor → Packet Scores → Cross-Stream Selector → Packet Data → Decoder

Model Inference

Video Frames
PacketGame Design
Contextual Predictor

- Available hint#2: packet-level metadata
  - Metadata: packet size & picture type
- Neural network-based predictor
- Training: offline collected pairs of (X, Y)

X: Packet Metadata

Y: Corresponding Redundancy Label
PacketGame Design
Contextual Predictor

• Packet scores returned by two modules, how to fuse them?
PacketGame Design
Contextual Predictor

- Fusing the temporal estimator’s output as another input view of NN
PacketGame Design
Contextual Predictor

- Fusing the temporal estimator’s output as another input view of NN
PacketGame Design
Contextual Predictor

- Fusing the temporal estimator’s output as another input view of NN
PacketGame Design
Cross-Stream Selector

- Combinatorial optimization problem

- Given predicted packet scores and packet decoding costs, under a decoding budget, maximize the summed scores of selected packets
PacketGame Design

Cross-Stream Selector

- Combinatorial optimization problem

- Given predicted packet scores and packet decoding costs, under a decoding budget, maximize obtained packet scores

Packet decoding costs depends on picture types and codec configurations (e.g., GOP size)
Packet Game Design
Cross-Stream Selector

- Combinatorial optimization problem
  - Given predicted packet scores and packet decoding costs, under a decoding budget, maximize obtained packet scores
  - we formulate this problem as an approximately fractional knapsack and prove the approximation ratio of the greedy algorithm
    - $1 - \frac{c}{B}$, in practice, typically greater than 95%
PacketGame Design
Overview

- Overall performance guarantee
  - we prove the regret in $T$ rounds is at most $\tilde{O}(\sqrt{T})$
Outline

• Background
• PacketGame Design
• Evaluation
Evaluation

Setting

- 2 public datasets, 1 dataset of collected from campus IP cameras, 3 types of sources
- 4 video inference tasks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Video Source</th>
<th>Inference Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Campus1K</td>
<td>IP Camera</td>
<td>Person Counting (PC)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Anomaly Detection (AD)</td>
</tr>
<tr>
<td>YT-UGC</td>
<td>Offline Video</td>
<td>Super-resolution (SR)</td>
</tr>
<tr>
<td>FireNet</td>
<td>Mobile Camera</td>
<td>Fire Detection (FD)</td>
</tr>
</tbody>
</table>

- opensource: [https://github.com/yuanmu97/PacketGame](https://github.com/yuanmu97/PacketGame)
Evaluation
Overall Performance

- target accuracy 90%, PacketGame achieves **2.1-4.8x end-to-end concurrency**
Evaluation
Ablation Study

- Contributions of contextual predictor and temporal estimator varies in different tasks

**Person Counting**

- Inference Accuracy

**Super-Resolution**

- Inference Accuracy

- Time Segments

- Contextual  Temporal  PacketGame  Random
Evaluation
Microbenchmarks

- PacketGame shows robust effectiveness with respect to involved variables, including training size
Evaluation
Microbenchmarks

- PacketGame shows robust effectiveness with respect to involved variables, including training size, window length

![Training Size](image1)

![Window Length](image2)
Evaluation
Microbenchmarks

- PacketGame shows robust effectiveness with respect to involved variables, including training size, window length, video codec, etc.

![Training Size Diagram]

![Window Length Diagram]

![Video Codec Diagram]
Evaluation
Microbenchmarks

- PacketGame shows robust effectiveness with respect to involved variables, including training size, window length, video codec, etc.

for more about design, theoretical analysis, experimental details, please read our paper :)

- Training Size
- Window Length
- Video Codec
Conclusions

Take-Home Messages

• The system bottleneck for multi-module pipeline is constantly changing, and now it’s the decoder’s turn for large-scale video analytics.

• Packet gating is promising and easy to implement. Try PacketGame for your video analytics system :)

• In the future, similar ideas could be explored for packet-level selection of other modalities, like audio and motion signals. Hope to inspire your research!
Acknowledgement

• My advisors Prof. Xiang-Yang Li and Prof. Lan Zhang in LINKE lab.

• Researchers and engineers at IAI for developing our video analytics system at USTC.

• Xuanke You, Shanyang Jiang, Miao-Hui Song, Changhu Can, Yueting Liu, Qing Chu, Ke Ding, Jin Yan
Thanks!

Q&A

Mu Yuan
Email: ymo813@mail.ustc.edu.cn