ExecutiveProducer(EP): Efficient ML Workflow Serving at the Edge for Video Analytics

Alexander Du, Ajay Krishnamurthy, Aining Liu
Department of Computer Science, Duke University

Duke

Summary

Video analytics applications involve

- Complex ML workflows with many interconnected models
- High bandwidth sensing and distributed edge-cloud computing

Operatoring such workflows poses numerous questions. Which models should be used? Where should the models be run? What leads to the lowest cost deployment?

***ExecutionProducer(EP)*: A system for optimizing and deploying video analytics workflows across edge and cloud**

- Identify best deployment plan at runtime
- Introduce video analytics specific tuning into search space

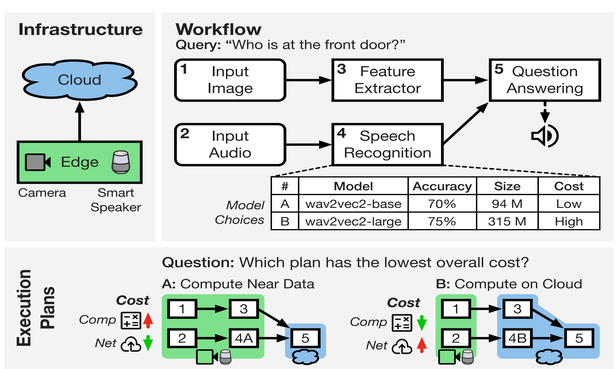
EP addresses complexity in deploying real-world video analytics applications

Motivation

- Plethora of model variants for any given task
- Edge and cloud have different cost and performance tradeoffs
- Choosing the best deployment plan requires joint consideration of both model choices and worker assignment

Determining effective deployment plans for a video analytic workflow is challenging because

a. the configuration search space is exponentially large
b. the optimal configuration depends on users' desired accuracy and cost targets
c. input video contents may exercise different paths in the workflow graph and produce variable intermediate results

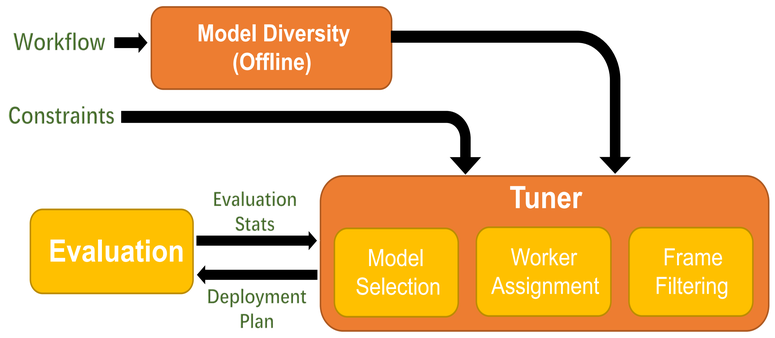


Large search space provides opportunity to explore cost and performance tradeoffs

Design

Inputs
- An arbitrary ML workflow
- ML model choices
- Infrastructure constraints
- Target accuracy

The system
- Determine deployment plan
- Satisfy accuracy constraint
- Reduce bandwidth and latency



Search space:
- *Frame filtering:* Alter the frame rate and resolution to reduce the amount of processing required downstream
- *Model selection:* Given a ML operator in the workflow, choose a model among the available variants that satisfies the accuracy constraint with the lowest cost
- *Worker assignment:* Determine the best mapping from models to available infrastructure workers that minimizes latency and bandwidth

DDS [1]
- Encodes significant areas of low quality frames with higher quality
- Saves bandwidth by encoding regions rather than entire frames

Model Diversity
- Compress models to tradeoff accuracy and efficiency
- Generate frontier of model choices to expand search space

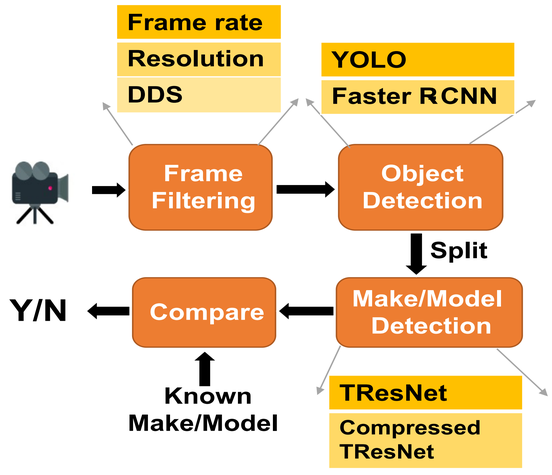
Search algorithm:
- Chameleon [2] observed configuration knobs independently impact accuracy
- This avoids an exponential search
- We use brute-force on the reduced search space

Simulation
- We use Python to simulate EP
- We compare the most accurate configuration with the current deployment plan for accuracy, latency, and bandwidth metrics

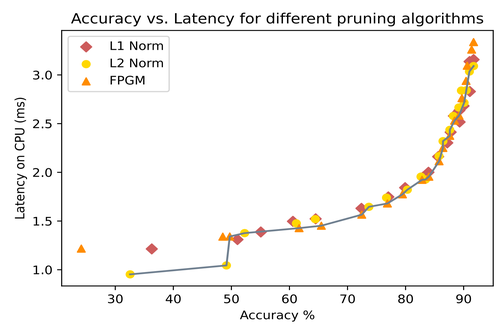
Automate search for the optimal deployment plan

Application Use Case

Use Case: AMBER Alert
- Leverage city cameras to locate specific cars.
- Workflow uses object detection with type-specific matching



Evaluation



Optimal Configuration: 3 fps, 384x216

Most Accurate Configuration: 15 fps, 1280x720

We achieve 93.7% bandwidth and 82.6% latency reductions while still identifying the target vehicle

References

- [1] Du, Kuntai and Pervaiz, Ahsan and Yuan, Xin and Chowdhery, Aakanksha and Zhang, Qizheng and Hoffmann, Henry and Jiang, Junchen, "Server-Driven Video Streaming for Deep Learning Inference," Association for Computing Machinery.
- [2] Jiang, Junchen and Ananthanarayanan, Ganesh and Bodik, Peter and Sen, Siddhartha and Stoica, Ion, "Chameleon: Scalable Adaptation of Video Analytics," Association for Computing Machinery.
- [3] Microsoft. Neural Network Intelligence (version v2.8).
- [4] T. Ridnik, H. Lawen, A. Noy, E. Ben, B. G. Sharir and I. Friedman, "TResNet: High Performance GPU-Dedicated Architecture," 2021 IEEE Winter Conference on Applications of Computer Vision (WACV).