Real-time deep learning on video streams

by Eran Avidan
Senior software engineer
Advanced Analytics @ Intel

- **Value**
  - Embed learning: Improve products power & performance
  - HW validation: Cut product time to market
  - Product Dev: Reduce test costs and improve quality
  - Sales: Increase revenue
  - Industrial AI: Reduce Manufacturing costs
  - Health: Improve clinical trials outcome

- **HOW**
  - Embed learning: Adaptive & personalized HW
  - HW validation: Automated validation with context
  - Product Dev: For every unit test only what's needed
  - Sales: Autonomous accounts coverage
  - Industrial AI: Proactive actuation to changes
  - Health: Continuance monitoring at home

**Vision:** Put AI to work for human experts
Video analysis
Video analysis for video understanding

Core challenges:

- Segmentation (pixel)
- Skeleton (clip)
- Detection & Recognition (region)
- Video captioning

Efforts in making models lighter and embed them on edge devices
Automating visual inspection
Video analysis for Visual inspections

Different defects may require different models

Anomaly behaviors lacks smoothness over time

Unique problems

Using available equipment for analysis
Centrally serving deep learning models with focus on low latency
For instance
Using the **lowest possible** inference time of each model
Challenges
Real time prediction serving challenges

Serving complex models (CNN) is part of the critical path

Fast turnaround

Batching to maximize throughput increases latency

Variety of models and frameworks

Scalable: when to scale?
Archelon

A scalable, fault tolerant and fully asynchronous serving system for video streams
The Application
Asynchronous Inference Unit

Always ON

Constant resource utilization

Dynamic Batching

Stateless

Scalable

Pre-Post processing

Data store

AIU

Model
Additional benefits
Separation of concerns

AIU

Data store

Resnet

PyTorch

Object detection

AIU
Better resource utilization

Process of multiple streams

Batch to reduce latency
When to scale?
Scaling criteria

Resource utilization

Latency threshold
The Data Store
Data store

Enables synchronization

Queue management platform

Supports single frames as well as ordered sequences of frames
The Model
Serving a Tensorflow model

Dedicated client

Protobufs
Model service

Common API

Generic client

Multiple protocols
Data flow
General suggestions to reduce latency

Receive a smaller image if possible

If not, resize the image on arrival

Separate presentation from prediction

Choose processing resources according to need

Implement the best serving method for each model
exit(0)
Real-time deep learning on video streams

by Eran Avidan
Senior software engineer